Fitting Finite Mixtures of Multivariate t-distributions via the EM Algorithm.

by

Mulenga Mwenya Francis

A dissertation submitted to the University of Zambia in partial fulfilment of the requirements for the degree of Master of Science in Statistics

THE UNIVERSITY OF ZAMBIA
LUSAKA

2022

Abstract

Finite mixture models are an important tool in modelling the distribution of a wide class of statistical problems. Mathematically, a $g-$component finite mixture model of multivariate t-distributions is specified by a convex combination (mixture distribution) of g multivariate t-distributions, which is given by

$$\mathcal{F}(\boldsymbol{x};\ \boldsymbol{\Psi}) = \sum_{j=1}^{g} \tau_j f(\boldsymbol{x};\ \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \nu_j)$$

where $\boldsymbol{\Psi} = \left(\tau_1, \cdots, \tau_{g-1}, \boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_g^T\right)^T$ is the parameter vector containing all the parameters of the mixture model, $\boldsymbol{\theta}_j = \left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \nu_j\right)^T$ is the parameter of the j^{th} component of the mixture model whose location parameter is $\boldsymbol{\mu}_j$, scale matrix $\boldsymbol{\Sigma}_j$, degrees of freedom ν_j and the j^{th} component mixing proportion τ_j, with $\sum_{j=1}^{g} \tau_j = 1$ for $j = 1, 2, \cdots, g$. In the computation of maximum likelihood estimates of $\boldsymbol{\Psi}$, the EM algorithm often outshines other iterative estimation procedures. The main shortcoming of the EM process is that convergence to the global mode is not guaranteed due to dependence on the starting point $\boldsymbol{\Psi}^{(0)}$. To address the problem of convergence to a local mode when fitting data to mixture models via the EM algorithm, optimized EM initialization methods such as k-means algorithm have been developed in the selection of $\boldsymbol{\Psi}^{(0)}$, especially when the underlying mixture model features Gaussian distributions. Compared to mixtures of Gaussian distributions however, mixtures of t-distributions have been identified as more robust modelling tools due to their heavier tails. The pitfall is that the later have more parameters to be estimated. This may make common EM initialization methods insufficient at attaining the global mode. Hence the need for more refined initialization methods. In this study, we extend the application of the burn-in techniques to mixtures of multivariate t-distributions. The performance of the burn-in scheme is compared with k-means algorithm, hierarchical clustering and random start based methods. The implementation of the EM algorithm using the package `EMMIXskew` in the statistical software **R**, shows that global convergence percentage is highest with the burn-in scheme initialized EM algorithm. With examples using various data sets, we show that the burn-in scheme is a competitive EM initialization method, even when the underlying mixture model features multivariate t-distributions.

Key Words: *Finite Mixture models, Mixture distributions, EM algorithm, Multivariate t-distributions, the k-means algorithm, Hierarchical clustering, Random starts, Burn-in scheme,* ***EMMIXskew****.*

Copyright

Approval

This M.Sc dissertation of Mulenga Mwenya Francis bearing the title: **Fitting Finite Mixtures of Multivariate t-distributions via the EM Algorithm**, has been approved as fulfilling the requirements or partial requirements for the award of Master of Science Degree in Statistics by the University of Zambia.

Name of Examiner 1: .....

Signature:.....

Date:.....

Name of Examiner 2: .....

Signature:.....

Date:.....

Name of Examiner 3: .....

Signature:.....

Date:.....

Chairperson, Board of Examiners: .....

Signature:.....

Date:.....

Declaration

The work described in this M.Sc dissertation was carried out under the supervision of Dr. Nawa M.V., Department of Mathematics and Statistics, The University of Zambia.

The M.Sc dissertation represents the original work by the author and has not otherwise been submitted in any form for any degree or diploma to any other University. Where use has been made of the work of others, it is duly acknowledged in the text.

Signed:

.....

Mulenga Mwenya Francis (Student)

Signed:

.....

Dr. Nawa M.V. (Supervisor)

Dedication

I dedicate this work to my mother Grace Chileshe, my brothers Kapembwa and Chipasha, my sisters Priscilla and Chishimba and my supportive and caring uncle, Chrispin Sampa Chileshe. A special dedication to the memory of my father, Green Mulenga.

Acknowledgements

I would like to express my heartfelt gratitude to Jehovah God for granting me good health without which the completion of my studies would not have been possible.

I would like to thank my supervisor, Dr. Nawa M.V., for the excellent supervision, guidance and tireless effort he invested in making sure that I reach the completion of my studies. His many comments, corrections, suggestions and many hours of discussion throughout this study made the completion of this work possible.

I would like to thank the Department of Mathematics and Statistics at the University of Zambia for granting me the opportunity to pursue my studies in a good and supportive environment.

I would also like to thank the Eastern African Universities Mathematics Programme (EAUMP) for the financial support rendered to me during my studies.

Last but not the least, I thank my family for the patience, support and encouragement I received throughout the duration of my studies. Special thanks to my uncle, Chrispin Chileshe for always encouraging me to pursue my studies and supporting me financially throughout my studies.

Contents

List of Tables

List of Figures

Notations and Conventions

$\boldsymbol{X}^p$	A p-dimensional random vector of p random variables, i.e, $\boldsymbol{X}^p = (X_1, \cdots, X_p)^T$
$\boldsymbol{x}^p$	An observed p-dimensional vector which is a specific realization of the vector $\boldsymbol{X}^p$, thus, $\boldsymbol{x}^p = (x_1, \cdots, x_p)^T$
$\boldsymbol{X}_o$	$n \times p$ random data matrix. A compact representation of the incomplete data random vectors, i.e, $\boldsymbol{X}_o = (\boldsymbol{X}_1^p, \cdots, \boldsymbol{X}_n^p)^T$.
$\boldsymbol{x}_o$	Specific realization of $\boldsymbol{X}_o$. A compact representation of actual observed (incomplete) data, thus, $\boldsymbol{x}_o = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$.
$\boldsymbol{Z}_i$	A g-dimensional random vector of latent variables for the i^{th} random vector $\boldsymbol{X}_i$, thus, $\boldsymbol{Z}_i = (Z_{i1}, \cdots, Z_{ig})^T$
$\boldsymbol{z}_i$	A g-dimensional latent vector of indicator scalars z_{ij}, for the i^{th} observed data point $\boldsymbol{x}_i$, thus, $\boldsymbol{z}_i = (z_{i1}, \cdots, z_{ig})^T$
$\boldsymbol{Z}$	An $n \times g$ random matrix of latent variables. A compact representation of indicator variables, $\boldsymbol{Z} = \left(\boldsymbol{Z}_1^T, \cdots, \boldsymbol{Z}_n^T\right)^T$
$\boldsymbol{z}$	A specific realization of the matrix $\boldsymbol{Z}$, i.e $\boldsymbol{z} = \left(\boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T\right)^T$
W_i	Random weight for the random data vector $\boldsymbol{X}_i$
w_i	A specific realization of W_i. A weight for data point $\boldsymbol{x}_i$
$\boldsymbol{X}_{pc}$	Compact representation of the partially complete random data matrix, $\boldsymbol{X}_{pc} = \left(\boldsymbol{X}_1^T, \cdots, \boldsymbol{X}_n^T, \boldsymbol{Z}_1^T, \cdots, \boldsymbol{Z}_n^T\right)^T$.
$\boldsymbol{x}_{pc}$	A specific value of the partially complete random data vector $\boldsymbol{X}_{pc}$ i.e, $\boldsymbol{x}_{pc} = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T, \boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T\right)^T$.
$\boldsymbol{X}_c$	A Compact representation of the complete random data vector, $\boldsymbol{X}_c = \left(\boldsymbol{X}_1^T, \cdots, \boldsymbol{X}_n^T, \boldsymbol{Z}_1^T, \cdots, \boldsymbol{Z}_n^T, W_1, \cdots, W_n\right)^T$.
$\boldsymbol{x}_c$	A compact representation of the complete sample data vector, $\boldsymbol{x}_c = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T, \boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T, w_1, \cdots, w_n\right)^T$.
$\boldsymbol{\Psi}$	Vector of all the parameters for the mixture model.
$\boldsymbol{\theta}_j$	Vector of all the parameters for the j^{th} component
$\boldsymbol{\mu}_j$	A $p \times 1$ mean vector for the j^{th} component
$\boldsymbol{\Sigma}_j$	A $p \times p$ scale matrix for the j^{th} component
ν_j	Degrees of freedom parameter for the j^{th} component
$\boldsymbol{\Psi}^{(m)}$	Value of $\boldsymbol{\Psi}$ at the m^{th} iteration of the EM algorithm.
$\boldsymbol{\Omega}$	Parameter space for the parametric family of multivariate t-distributions so that $\boldsymbol{\theta}_j \in \boldsymbol{\Omega} \ \ \forall\ j, \ \ j = 1, \cdots, g$
$\overline{\boldsymbol{\Omega}}$	Parameter space for the mixture model so that $\boldsymbol{\Psi} \in \overline{\boldsymbol{\Omega}}$
$\mathbf{t}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$	A p-variate t-distribution with center parameter $\boldsymbol{\mu}$, scale matrix parameter $\boldsymbol{\Sigma}$ and degrees of freedom parameter ν

CHAPTER 1
Introduction

This chapter presents an introduction to this dissertation. Much of the material presented in this chapter will be detailed considered in the proceeding chapters. Here, we present a background and overview of our study.

1.1. An Introduction to Finite Mixture Models

Let $f_1(x;\ \theta_1), \cdots, f_g(x;\ \theta_g)$ be a finite collection of g arbitrary probability density functions. A $g-$component finite mixture probability density function is given by

$$f(x : \mathbf{\Psi}) = \sum_{j=1}^{g} \tau_j f_j(x; \theta_j) \tag{1.1}$$

where $\mathbf{\Psi} = (\tau_1, \cdots, \tau_{g-1},\ \theta_1, \cdots, \theta_g)^T$ is the vector containing all the parameters of the mixture distribution, θ_j is the parameter vector for the j^{th} component and τ_j is the mixing proportion for the j^{th} component with $\tau_j > 0$ and $\sum_{j=1}^{g} \tau_j = 1$. A probability model described or specified by a finite mixture probability density function is what we call a finite mixture model. If the mixture distribution has g component distributions, then we have a g-component finite mixture model. If a random variable X follows a finite mixture distribution in (1.1), then X is called a finite mixture random variable. Similarly, a p-variate vector $\boldsymbol{X}^p$ that follows a multivariate mixture distribution is a p-variate mixture random vector.

From this definition, we take note that a finite sum of g probability density functions is a probability density function for some g-component mixture probability distribution only if the sum is a convex combination of the component density functions [1]. In principle, the component distributions f_j in the finite mixture model (1.1) may all be from different distributions. In practice, a lot of attention is given to parametric mixture models where the component distributions are all from the same parametric family, say $\{f(x,\ \theta_j)|\ \theta_j \in \Omega$ where Ω is the parameter space $\}$, but with different parameters $\theta_j \in \Omega$. In this case, the model in (1.1) maybe written as

$$\mathcal{F}(x : \mathbf{\Psi}) = \sum_{j=1}^{g} \tau_j f(x; \theta_j) \tag{1.2}$$

where $\mathcal{F}(.)$ and $f(.)$ denote the mixture density function and the density function for the parametric family, respectively. The main goal in finite mixture modelling is to draw inferences about the mixture model parameter $\boldsymbol{\Psi}$, which is often estimated by its maximum likelihood (ML) estimate $\hat{\boldsymbol{\Psi}}$.

The standard tool used to obtain ML estimates of the mixture model parameter $\boldsymbol{\Psi}$ is the Expectation Maximization (EM) algorithm [1][2]. The EM algorithm is an iterative procedure for computing ML estimates in cases with missing data or hidden (latent) variables. Briefly, if $\boldsymbol{x}_o = (x_1, \cdots, x_n)$ denotes an observed sample data of size n, then with a mixture model-based approach to drawing inferences from the observed data set $\boldsymbol{x}_o$, each data point x_i is assumed to be a realization of the mixture random variable X with probability density function as defined in (1.2). We use the observed data set $\boldsymbol{x}_o$ to compute the ML estimates for the model parameter $\boldsymbol{\Psi}$ of the finite mixture model. In the EM framework, $\boldsymbol{x}_o$ is considered to be an incomplete data set until the component indicator variables (missing data) $\boldsymbol{Z} = \left(\boldsymbol{Z}_1^T, \cdots, \boldsymbol{Z}_n^T\right)^T$ are introduced into the model where the g-dimensional latent random vector $\boldsymbol{Z}_i = (Z_{i1}, \cdots, Z_{ig})^T$ has entries defined as

$$Z_{ij} = \begin{cases} 1 & \text{if } x_i \text{ belongs to the } j^{th} \text{ component} \\ \\ 0 & \text{otherwise .} \end{cases}$$

A realization of the random latent vector $\boldsymbol{Z}_i$ is a vector $\boldsymbol{z}_i = (z_{i1}, \cdots, z_{ig})^T$ which defines the component of origin of the data point x_i accordingly as, $z_{ij} = 1$ if observation x_i belongs to the j^{th} component and $z_{ij} = 0$ otherwise [2]. The complete data $\boldsymbol{x}_c$ is now specified as

$$\boldsymbol{x}_c = \left(x_1, \cdots, x_n,\ \boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T\right)^T \tag{1.3}$$

It can be shown (see [2],[4] and Section 3.2 of this dissertation) that the corresponding log-likelihood function $l_c(\boldsymbol{\Psi}, \boldsymbol{z})$, called the complete data log-likelihood function is given by

$$l_c(\boldsymbol{\Psi}, \boldsymbol{z}) = \sum_{i=1}^{n} \sum_{j=1}^{g} z_{ij} \left[\ln \tau_j + \ln f(x;\ \theta_j)\right] \tag{1.4}$$

Using the function $l_c(\boldsymbol{\Psi}, \boldsymbol{z})$ in (1.4), the EM algorithm generates a sequence of parameter estimates $\{\boldsymbol{\Psi}^{(m)}\}_{m=0}$ iteratively until the sequence converges to some optimum value. The EM algorithm starts from a preselected starting point $\boldsymbol{\Psi}^{(0)}$, which is the initial value of $\boldsymbol{\Psi}$ when the iteration time is $m = 0$, and proceeds to generate the values of the sequence $\{\boldsymbol{\Psi}^{(m)}\}_{m=0}$ iteratively. Each iteration of the EM algorithm consists of two steps; the Expectation step (E-step) and the Maximization step (M-step). In the E-step at the $(m+1)^{th}$ iteration, the conditional expected value of the complete data log-likelihood function in (1.4), is evaluated with respect to the observed data $\boldsymbol{x}_o$ and the current parameter estimate, $\boldsymbol{\Psi}^{(m)}$. Thus, defining $Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)})$ as the expected value of the complete data log-likelihood function of $\boldsymbol{\Psi}$ with respect to the current conditional distribution of $\boldsymbol{Z}$ given the

observed data $\boldsymbol{x}_o$ and the current estimates of the parameters $\boldsymbol{\Psi}^{(m)}$, then E-step of the EM algorithm computes

$$Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)}) = E_{\boldsymbol{Z}|\boldsymbol{x}_o,\boldsymbol{\Psi}^{(m)}}\left[l_c(\boldsymbol{\Psi}, \boldsymbol{Z})\right] \tag{1.5}$$

This step gives an estimate of the latent variables $\boldsymbol{Z}_i$ $\forall i = 1, \cdots, n$, using the observed data $\boldsymbol{x}_0$ and the current value $\boldsymbol{\Psi}^{(m)}$ of the model parameter $\boldsymbol{\Psi}$. In the M-step, the expected complete data log-likelihood function in (1.5) is maximized to produce $\boldsymbol{\Psi}^{(m+1)}$, the updated value of the model parameter $\boldsymbol{\Psi}$. This step computes

$$\boldsymbol{\Psi}^{(m+1)} = \arg\max_{\boldsymbol{\Psi}} \; Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)}) \tag{1.6}$$

In the M-step, the likelihood function is maximized under the assumption that the values of the latent random variables $\boldsymbol{Z}_i$ are known. The estimated value $\boldsymbol{z}$ from the E-step is used in lieu of the actual missing data $\boldsymbol{Z}$. The EM algorithm iterates between the E-step and the M-step until convergence. The algorithm is said to have converged when the change in the log-likelihood value is sufficiently small.

1.2. Statement of the problem

A finite mixture model that uses the Student's t-distribution has been recognised as a more robust extension of normal mixtures [7]. The main pitfall however, is that the use of t-distributions increases the number of parameters to be estimated compared to the use of Gaussian distributions. This may make some of the common EM initialization methods insufficient at attaining the global mode. The EM algorithm may fail to converge to the global mode when the starting point is poorly chosen [1]. Hence the need to optimize initialization methods for obtaining suitable starting values of $\boldsymbol{\Psi}$. This study analyses four initialization methods for selecting starting values of $\boldsymbol{\Psi}$, when the underlying model features multivariate t-distributions with the aim of determining an optimal initialization method.

1.3. Aim of the Study:

This study aims to determine an optimal method of parameter initialization when fitting data to finite mixture models of multivariate t-distributions via the Expectation Maximization algorithm.

1.4. Research Objectives and Hypothesis Questions

This study focuses on finite mixture models with components that feature multivariate t-distributions. We review three common methods used in the selection of $\boldsymbol{\Psi}^{(0)}$, the starting point for the EM algorithm, when fitting data sets to mixture models that feature components with multivariate

t-distributions; the k-means algorithm, the hierarchical clustering and random start methods [10][11][21]. Further, we investigate an alternative method, called the burn-in scheme [4][5]. In summary, this study archives the following study objectives:

(i) Review the techniques namely random start, k-means, hierarchical clustering and burn-in scheme for initialization of the EM algorithm to find ML estimates of parameters in finite mixtures of multivariate gaussian distributions.

(ii) To extend the techniques of random start, k-means, hierarchical clustering and burn-in scheme for initialization of the EM algorithm to find ML estimates of parameters in finite mixtures of multivariate t-distributions.

(iii) To compare efficiencies of the random start, k-means, hierarchical clustering and the burn-in scheme when fitting data to finite mixtures of multivariate t-distributions by fitting sample data.

With our outlined research procedure above, we provide answers to the following critical research questions:

(i) When fitting data to finite mixtures of multivariate t-distributions via EM algorithm, does the burn-in scheme yield competitive solutions as compared to the dominant methods such as random starts, k-means and hierarchical clustering?

(ii) When fitting data to finite mixtures of multivariate t-distributions via the EM algorithm, Which initialization method gives the most optimal solution?

1.5. Significance of the Study

When the underlying data set $\boldsymbol{x}_o$ is of continuous nature, the most widely used parametric family in mixture modelling is the normal (Gaussian) family of distributions due to their computational convenience [4][5][6]. However, for many applied problems, the tails of the Gaussian distribution are often shorter than required. Thus, the estimates of the component means and covariance matrices can be affected by observations that are atypical of the components in the normal mixture model being fitted. The use of the t-distribution to model the component distributions in mixture modelling has an advantage over the use of the Gaussian distribution in that the procedure gives a more robust approach because the degrees of robustness can be inferred from the given data by computing the ML estimate of the degrees of freedom parameter present in the t-distribution [3][8]. Hence the need to study finite mixture models that feature components that follow the t-distribution.

1.6. Literature Review

In the computation of ML estimates of parameters in finite mixture models, the EM algorithm often outperforms other iterative procedures such as the Newton-Raphson and the gradient descent methods [1]. In its general form, the EM algorithm is fairly easy to implement as it requires neither the computation of the score nor information functions, unlike the Newton-Raphson method and gradient descent based methods. Further, the EM algorithm has an advantage in that each iteration always increases the log-likelihood function [2]. Thus, the EM algorithm is guaranteed to drive the likelihood function towards a mode, whether local or global [13]. The main shortcoming of the EM algorithm is that when the likelihood function is multi-modal, convergence to the global mode is not guaranteed [1]. Convergence can either be to an optimal solution or an inferior solution depending on the selected starting point $\boldsymbol{\Psi}^{(0)}$ of the algorithm [13]. Thus, in some instances, the concept of convergence can just be an indication of lack of progress in the process rather than true convergence [4]. This will be the case when the process is trapped at some local mode of the log-likelihood function. Quite often, convergence to a local maximum will occur when the selected starting point is in the vicinity of that local mode. Therefore, to avoid convergence to a local mode and increase the chances of convergence to the optimum mode, the selection process of the starting point for the EM algorithm must be optimized.

To address the problem of convergence to a local mode in the EM algorithm, a number of modifications to the General Expectation Maximization (GEM) algorithm of [2] have been made to produce EM variants (See [1]). These EM variants or EM extensions have modified the original general form of the EM algorithm in an attempt to improve its algorithmic efficiency. Some of the notable variants include; the Expectation Conditional Maximization (ECM) algorithm [15], the Expectation Conditional Maximization Either (ECME) algorithm [16], the emEM algorithm [5], the Multi-cycle EM algorithm [15], the Classification Expectation Maximization (ECM) algorithm [5], the Sparse Expectation Maximization (SEM) algorithm [9], the Moment Matching EM [6] and the Stochastic Expectation Maximization algorithm [5] among others. Some of these variants have combined the techniques of other iterative algorithms such as the Newton-Raphson methods and GEM in order to improve the overall algorithmic performance [17][18]. It has been noted however, that implementing some of these EM variants can be computationally burdensome as most of them have altered the simplicity of the GEM [2][4]. Therefore, optimizing the initial value selection process while preserving the simplicity of the GEM algorithm has been considered as a more promising approach of optimizing the algorithmic efficiency when fitting observed data sets to mixture models [4][5][6].

Generally, mixture models that feature non-Gaussian components such as t-distributions, skew-normal distributions or skew-t distributions can be used to effect better clustering solutions in situations where the data set is either heterogeneous, contain outliers, is asymmetric or has non-elliptical groups [10][11][12]. These models however, have more parameters to be

estimated than mixtures of Gaussian distributions. Their use in mixture modelling results in more complex models [4][12]. For mixtures models featuring t components, the k-means algorithm, hierarchical clustering and random start approach are some of the dominant methods of parameter initialization [11][21]. In an attempt to attain better starting values for the EM algorithm when modelling the data using mixtures of Gaussian distributions, [4] introduced the burn-in function as a means of parameter initialization. The concepts in the burn-in scheme have been suggested to have a promising application in other more complex distributions such as mixtures of skew normal, mixtures of t, mixtures of skew t and many other complex distributions [4][22]. In this study, we focus on the dominant parameter initialization methods; the k-means algorithm, hierarchical clustering and random starts, when modelling observed data sets using mixtures of multivariate t-distributions via the EM algorithm. Further, we extend the concepts of the burn-in scheme to mixtures of multivariate t-distributions.

1.7. Research Methodology

Let $\mathbf{t}_p\left(\boldsymbol{\mu},\ \boldsymbol{\Sigma},\ \nu\right)$ denote a multivariate (p-variate) t-distribution with location parameter $\boldsymbol{\mu}$, scale parameter $\boldsymbol{\Sigma}$ and degrees of freedom parameter ν. Further, let $\left\{ f(\boldsymbol{x};\ \boldsymbol{\theta}) \,\middle|\, \boldsymbol{\theta} = (\boldsymbol{\mu},\ \boldsymbol{\Sigma},\ \nu)^T \in \boldsymbol{\Omega} \right\}$ denote the parametric family of multivariate t-distributions with parameter space $\boldsymbol{\Omega}$. If $\boldsymbol{X}_1^p, \cdots, \boldsymbol{X}_g^p$ is a class of $p-$dimensional random vectors such that $\boldsymbol{X}_j^p \ \sim t_p\left(\boldsymbol{\mu}_j,\ \boldsymbol{\Sigma}_j,\ \nu_j\right)$ for some $\boldsymbol{\theta}_j = \left(\boldsymbol{\mu}_j,\ \boldsymbol{\Sigma}_j,\ \nu_j\right)^T \in \boldsymbol{\Omega}$ with $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j$ if $i \neq j$ for $j, i = 1, \cdots, g$, then a convex combination of the g density functions $f(\boldsymbol{x};\ \boldsymbol{\theta}_j)$ denoted by $\mathcal{F}(\boldsymbol{x};\ \boldsymbol{\Psi})$, is a $g-$component finite mixture distribution of multivariate t-distributions if

$$\mathcal{F}(\boldsymbol{x};\ \boldsymbol{\Psi}) = \sum_{j=1}^{g} \tau_j f(\boldsymbol{x};\ \boldsymbol{\theta}_j) = \sum_{j=1}^{g} \tau_j f(\boldsymbol{x};\ \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \nu_j) \tag{1.7}$$

where $\boldsymbol{\Psi} = \left(\tau_1, \cdots, \tau_{g-1}, \boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_g^T\right)^T$ is the vector containing all the parameters of the mixture model, $\boldsymbol{\theta}_j = \left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \nu_j\right)^T$ contains all the parameters for the j^{th} component of the model and $\tau_j > 0$ is the mixing proportion for the j^{th} component of the model with $\sum_{j=1}^{g} \tau_j = 1$, for $j = 1, 2, \cdots, g$. Let $\boldsymbol{x}_o = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$ denote an observed p-dimensional sample data of size n which we wish to fit to a g-component mixture model of multivariate t-distributions in (1.7). Each observed data point $\boldsymbol{x}_i$ in $\boldsymbol{x}_o$ for $i = 1, \cdots, n$ is assumed to be a realization of a mixture random vector $\boldsymbol{X}$ with probability density function (1.7). To fit a g-component mixture model using the observed data $\boldsymbol{x}_o$, we obtain the ML estimate of the model parameter $\boldsymbol{\Psi}$ via the EM algorithm. This requires selecting $\boldsymbol{\Psi}^{(0)}$, the starting point for the EM algorithm. For a mixture model in (1.7), the starting point is understood to be a vector $\boldsymbol{\Psi}^{(0)}$ given by

$$\boldsymbol{\Psi}^{(0)} = \left(\tau_1^{(0)}, \cdots, \tau_g^{(0)}, \boldsymbol{\theta}_1^{T(0)}, \cdots, \boldsymbol{\theta}_g^{T(0)}\right)^T \tag{1.8}$$

where $\boldsymbol{\theta}_j^{(0)} = \left(\boldsymbol{\mu}_j^{(0)T}, \boldsymbol{\Sigma}_j^{(0)}, \ \nu_j^{(0)}\right)^T$. We use the observed data set $\boldsymbol{x}_o$ to select the starting point $\boldsymbol{\Psi}^{(0)}$.

The observed data set $\boldsymbol{x}_o$ can be partitioned into g clusters so that $\boldsymbol{\Psi}^{(0)}$ is obtained from the resulting partition. For example, suppose that $\boldsymbol{S} = \{\boldsymbol{S}_1, \cdots, \boldsymbol{S}_g\}$ is a partition of the observed data set $\boldsymbol{x}_o$ into g clusters such that $\boldsymbol{x}_i \in \boldsymbol{S}_h$ and $\boldsymbol{x}_i \in \boldsymbol{S}_k$ implies $\boldsymbol{S}_h = \boldsymbol{S}_k$ for $\boldsymbol{S}_h, \boldsymbol{S}_k \in \boldsymbol{S}$. Then the initial estimate value of the j^{th} component parameters, $\tau_j^{(0)}$ and $\boldsymbol{\theta}_j^{(0)}$, can be set as

$$\boldsymbol{\mu}_j^{(0)} = \frac{1}{|\boldsymbol{S}_j|} \sum_{\boldsymbol{x}_i \in \boldsymbol{S}_j} \boldsymbol{x}_i \ , \qquad \boldsymbol{\Sigma}_j^{(0)} = \frac{1}{|\boldsymbol{S}_j|} \sum_{\boldsymbol{x}_i \in \boldsymbol{S}_j} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(0)}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(0)}\right)^T$$

$$\tau_j^{(0)} = \frac{|\boldsymbol{S}_j|}{n} \quad \text{and} \quad \nu_j^{(0)} = 4$$

for all $j = 1, \cdots, g$ (see [11] and [21]). Grouping of the observed data set $\boldsymbol{x}_o$ into g clusters occurs through one of the following initialization methods:

(a) **k-means algorithm:** Partition the observed data set $\boldsymbol{x}_o$ into k=g clusters, $\boldsymbol{S} = \{\boldsymbol{S}_j| \ \text{ for } \ j = 1, 2, \cdots, k\}$ so as to minimize the within-cluster sum of squares (WCSS). The k-means algorithm solves the equation

$$\arg\min_{\boldsymbol{S}} \sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in \boldsymbol{S}_j} d\left(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(0)}\right) = \arg\min_{\boldsymbol{S}} \sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in \boldsymbol{S}_j} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(0)}\|_2^2$$

where $\boldsymbol{\mu}_j^{(0)}$ is the mean (centroid) of cluster j, $\boldsymbol{S}_j$ is the set of all points in the j^{th} cluster and $d(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(0)})$ denotes the squared Euclidean distance of the point $\boldsymbol{x}_i$ to the centroid $\boldsymbol{\mu}_j^{(0)}$.

(b) **Hierarchical clustering:** Partition the observed data set $\boldsymbol{x}_o$ into g clusters, $\boldsymbol{S} = \{\boldsymbol{S}_j| \ \text{ for } \ j = 1, 2, \cdots, g\}$ so as to minimize the inter-cluster squared Euclidean distance defined using the maximum-linkage criterion:

$$D(\boldsymbol{S}_h, \ \boldsymbol{S}_k) = \max_{\boldsymbol{x}_h \in \boldsymbol{S}_h, \ \boldsymbol{x}_k \in \boldsymbol{S}_k} d(\boldsymbol{x}_h, \ \boldsymbol{x}_k) \qquad\qquad h, \ k = 1, 2, \cdots, g$$

where D is the distance between clusters, d is the metric used, in this case the squared Euclidean distance, $\boldsymbol{x}_k$ and $\boldsymbol{x}_h$ are data points belonging to clusters $\boldsymbol{S}_k$ and $\boldsymbol{S}_h$, respectively.

(c) **Random start:** Several random start approaches are in use. One method randomly groups the observations into g groups without regard to any metric and proceeds to computing $\boldsymbol{\Psi}^{(0)}$ from the resulting partition.

(d) **Burn-in scheme:** Recall that in the EM framework, $\boldsymbol{x}_o$ is considered incomplete until the indicator variables $\boldsymbol{Z} = \left(\boldsymbol{Z}_1^T, \cdots, \boldsymbol{Z}_n^T\right)^T$ are introduced into the model such that $\boldsymbol{Z}_i$ defines the component of origin of data point $\boldsymbol{x}_i$. A particular value of the random matrix

$\boldsymbol{Z} = \left(\boldsymbol{Z}_1^T, \cdots, \boldsymbol{Z}_n^T\right)^T$ say $\boldsymbol{z} = \left(\boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T\right)^T$, gives a unique partition of the observed sample data $\boldsymbol{x}_o$ into g groups of the mixture model. Given a partition $\boldsymbol{S} = \{\boldsymbol{S}_1, \cdots, \boldsymbol{S}_g\}$, we let $\boldsymbol{S}_j = \{\boldsymbol{x}_i |\ \boldsymbol{z}_{ij} = 1\}$ denote a set of all data points in the j^{th} component. Optimizing the choice of $\boldsymbol{\Psi}^{(0)}$ can be attained through optimizing the choice of the initial $\boldsymbol{Z}$ matrix as this would give an optimized choice of the initial partition of the observed data set.

Let $\boldsymbol{z}^{(0)}$ denote the initial value of the matrix $\boldsymbol{Z}$ and $\boldsymbol{S}^{(0)}$ denote the corresponding partition of the observed data set $\boldsymbol{x}_o$ into g clusters i.e $\boldsymbol{z}^{(0)} = (\boldsymbol{z}_1^{T(0)}, \cdots, \boldsymbol{z}_g^{T(0)})^T$ and $\boldsymbol{S}^{(0)} = \{\boldsymbol{S}_1^{(0)}, \cdots, \boldsymbol{S}_g^{(0)}\}$ denote the values of the $\boldsymbol{Z}$ matrix and the set of clusters $\boldsymbol{S}$, respectively, at iteration time $m = 0$. The main objective of the burn-in scheme is to generate an optimized value of the $\boldsymbol{Z}$ matrix by analyzing a number of candidate $\boldsymbol{Z}$ matrix values. The burn-in scheme is itself an iterative process consisting of the following simple steps:

(i) Generate 2^a candidate values of the $\boldsymbol{Z}$ matrix, $\boldsymbol{z}_1^{(0)}, \cdots, \boldsymbol{z}_{2^a}^{(0)}$ where $a \in \mathbb{N}$. In practice, $a = 4, 5, 6$.

(ii) For each matrix $\boldsymbol{z}_k^{(0)}$ where $k = 1, \cdots, 2^a$, obtain corresponding EM starting points $\boldsymbol{\Psi}_k^{(0)}$.

(iii) Conduct a single pair of EM steps for each candidate $\boldsymbol{z}_k^{(0)}$ using starting point $\boldsymbol{\Psi}_k^{(0)}$. Concurrently update the parameter estimates from $\boldsymbol{\Psi}_k^{(0)}$ to $\boldsymbol{\Psi}_k^{(1)}$ relating to each matrix $\boldsymbol{z}_k^{(0)}$.

(iv) Evaluate the log-likelihood $l\left(\boldsymbol{\Psi}_k^{(1)}, \boldsymbol{z}^{(0)} \middle| \boldsymbol{x}_o\right)$ relating to each $\boldsymbol{z}_k^{(0)}$.

(v) Rank the $\boldsymbol{z}_k^{(0)}$ matrices in descending order of their corresponding observed log-likelihood.

(vi) Discard the lower half of the $\boldsymbol{z}_k^{(0)}$ matrices based on the corresponding log-likelihood values.

(vii) Repeat steps (ii)—(vi) using the remaining candidate $\boldsymbol{z}_k^{(0)}$ matrices each time increasing the number of EM iterations by a multiplicative factor of 2, until only one $\boldsymbol{z}_k^{(0)}$ matrix remains.

The emerging matrix from this process is considered to be the optimized value of the $\boldsymbol{Z}$ matrix for use in the full EM algorithm.

The remainder of this dissertation is organized as follows:

Chapter 2: In this chapter, we recall the definition and some of the basic properties of the t-distribution for a single univariate random variable say X. Further, we extend the definition of a t-distribution to a p-variate random vector $\boldsymbol{X}^p$ as a multivariate case. Here, we recall the generalization of the univariate t-distribution to the multivariate t-distribution and present its relationship with the multivariate Gaussian and the univariate Gamma distributions [23][24]. The well known derivation of the multivariate t-distribution from the multivariate Gaussian distribution and the univariate Gamma distribution through the process of compounding is reviewed [19].

This chapter concludes with a brief discussion on the maximum likelihood estimation of parameters for mixtures of multivariate t-distributions, and outlines the challenges associated with the ML estimation process when the underlying model features a mixture distribution [1][2].

Chapter 3: In this chapter, we review the EM algorithm and the theory underlying the general EM algorithm [1][2]. The details of the E-Step and M-step of the EM algorithm are thoroughly reviewed as used in mixtures featuring multivariate t-distributions [3]. We review the convergence properties of the EM algorithm and recall some convergence criterions used to stop the EM process [13][14]. Further, this chapter discusses the parameter initialization methods for the EM algorithm. We review the random start initialization method [1][25], the k-means clustering algorithm [27] and the hierarchical clustering as parameter initialization methods in the EM algorithm [33]. Finally, we review the concepts of the burn-in scheme and present its extended application to mixtures featuring multivariate t components.

Chapter 4: In this chapter, we present the illustrative data sets used in our investigation of the research questions. Further, the experimental strategies used in our investigations are thoroughly outlined. The relevant packages and functions used in the statistical software **R** are presented and their use described.

Further, this chapter details the experimental processes and presents a summary of the results using figures and tables. The four methods of parameter initialization; random start, k-means, hierarchical clustering and burn-in scheme, are applied to each of the data sets and the respective convergent log-likelihood values are documented, detailing the performance of each method. In this chapter, we discuss and compare the results of our experimental findings. The performance of each initialization method is assessed and comparisons based on the average values of the convergent log-likelihoods, Bayesian information criterion, Akaike information criterion and convergence error are made. The promising performance of the burn-in scheme is detailed.

Chapter 5: This chapter summarizes the main findings of this study in relation to our hypotheses questions. As a concluding chapter, this chapter gives a summary of the main findings of this study. The answers to our research questions are summarized and further work that can be taken in line with this research is suggested.

CHAPTER 2
Mixture Models Featuring Multivariate t-distributions

In the first section of this chapter, we review the derivation of the generalized t-distribution from compounding the general Gaussian distribution with its variance distributed according to an inverse gamma distribution. This derivation shows that the t-distribution has the same symmetrical shape as a normal distribution with the same central point μ, but has greater variance and heavier tails. In section two, the derivation of the multivariate t-distribution from the multivariate Gaussian and the Gamma distributions using the concepts of compound distributions, is revised. We recall a mixture model of t-distributions and conclude this chapter with an introduction to ML estimation of parameters in mixtures of t-distributions.

2.1. Some Basic Properties of the t-distribution

Definition 2.1.1. *A continuous random variable T is said to have a standardized t-distribution with ν degrees of freedom if it has the probability density function;*

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\ \ \Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}} \tag{2.1}$$

where $t \in (-\infty,\ \infty)$ and $\nu \in (0,\ \infty)$. If T has the t-distribution with degrees of freedom ν, we will write $T \sim t_1(\nu)$.

Definition 2.1.2. *A continuous random variable Z follows the standard normal (Gaussian) distribution if it has the probability density function*

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \tag{2.2}$$

where $z \in (-\infty,\ \infty)$. If Z follows the standard Gaussian distribution, we write $Z \sim N(0, 1)$.

Definition 2.1.3 ([19]). *A continuous random variable W, is said to have a Gamma distribution with scale parameter $\lambda > 0$ and shape parameter $\alpha > 0$, if it has a probability density function of the form*

$$f(w) = \frac{\lambda^{\alpha}}{\Gamma(\alpha)}\ w^{\alpha-1}e^{-\lambda w} \qquad \text{where} \quad 0 < w < \infty \tag{2.3}$$

A special notation used to designate that W has a gamma distribution is $W \sim Gamma(\alpha, \lambda)$.

Definition 2.1.4. *Let W denote a continuous random variable that follows a Gamma distribution with scale parameter $\lambda = \frac{1}{2}$ and shape parameter $\alpha = \frac{\nu}{2}$ where $\nu \in \mathbb{Z}^+$. Then the probability density function of W is given by*

$$f(w) = \frac{1}{2^{\frac{\nu}{2}}\ \Gamma\left(\frac{\nu}{2}\right)} w^{\frac{\nu}{2}-1}e^{-\frac{1}{2}w} \tag{2.4}$$

and we say W follows a chi-square distribution with ν degrees of freedom. A special notation used to designate that W has a chi-square distribution is $W \sim \chi^2(\nu)$.

Theorem 2.1.5. *Let Z denote a random variable that follows the standard normal distribution and W a random variable that has a chi-square distribution with ν degrees of freedom. If Z and W are independent, then the random variable*

$$T = \frac{Z}{\sqrt{\frac{W}{\nu}}}$$

has a t-distribution with ν degrees of freedom.

Proof. The joint density function of the independent variables Z and W is given by the product of their respective marginal density functions. Thus,

$$f(w, z) = \frac{1}{2^{\frac{\nu}{2}}\sqrt{2\pi}\ \Gamma\left(\frac{\nu}{2}\right)} w^{\frac{\nu}{2}-1}e^{-\frac{1}{2}(z^2+w)}$$

Let $Y = W$ and $T = \frac{Z}{\sqrt{\frac{W}{\nu}}}$. Then $W = Y$ and $Z = T\sqrt{\frac{Y}{\nu}}$ so that the Jacobian of the transformation $(W,\ Z) \to (T,\ Y)$ is $J = -\sqrt{\frac{y}{\nu}}$. Then

$$\begin{aligned} f(t, y) =& \frac{1}{2^{\frac{\nu}{2}}\sqrt{2\pi}\ \Gamma\left(\frac{\nu}{2}\right)} y^{\frac{\nu}{2}-1}e^{-\frac{1}{2}\left[\left(t\sqrt{\frac{y}{\nu}}\right)^2+y\right]}.|J| \\ =& \frac{1}{2^{\frac{\nu}{2}}\sqrt{2\pi\nu}\ \Gamma\left(\frac{\nu}{2}\right)} y^{\frac{\nu+1}{2}-1}e^{-\frac{y}{2}\left(\frac{t^2}{\nu}+1\right)} \end{aligned}$$

which is the joint density function of T and Y. Integrating $f(t, y)$ over y gives the marginal density function of T. Thus,

$$\begin{aligned} f(t) =& \int_Y f(t, y)\ dy \\ =& \frac{1}{2^{\frac{\nu}{2}}\sqrt{2\pi\nu}\ \Gamma\left(\frac{\nu}{2}\right)} \int_0^{\infty} y^{\frac{\nu+1}{2}-1}e^{-\frac{y}{2}\left(\frac{t^2}{\nu}+1\right)}\ dy \end{aligned}$$

$$= \frac{1}{2^{\frac{\nu}{2}}\sqrt{2\pi\nu}\ \Gamma\left(\frac{\nu}{2}\right)}\ \mathcal{L}\left(y^{\frac{\nu+1}{2}-1}\right)$$
$$= \frac{1}{2^{\frac{\nu}{2}}\sqrt{2\pi\nu}\ \Gamma\left(\frac{\nu}{2}\right)}\ \frac{\left(\frac{\nu+1}{2}-1\right)!}{\left[\frac{1}{2}\left(\frac{t^2}{\nu}+1\right)\right]^{\frac{\nu+1}{2}}}$$

where $\mathcal{L}\left(h(.)\right)$ denotes the Laplace transform of the function $g(.)$. Thus,

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\ \ \Gamma\left(\frac{\nu}{2}\right)}\ \left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}} \tag{2.5}$$

which is the probability density function for the random variable T that follows the standard t-distribution with ν degrees of freedom. □

Remark 2.1.1. If a random variable T is such that $T \sim t_1(\nu)$, then

(i) $\mathbb{E}\left(T\right) = 0$ for $\nu \in (1,\ \infty)$, otherwise $\mathbb{E}\left(T\right)$ is undefined .

(ii) Mode $(T) = 0$

(iii) Median $(T) = 0$.

(iv) $\mathbb{E}\left[T - \mathbb{E}\left(T\right)\right]^2 = \frac{\nu}{\nu-2}$ for $\nu > 2$,

The classic t-distribution discussed so far is the standard t-distribution. By introducing a location parameter μ and a scale parameter σ, the classic t-distribution can be generalized to a three parameter location-scale family

$$T = \frac{X-\mu}{\sigma}$$

so that the density function for the generalized (non-standardized) t-distribution is written as

$$f(x;\ \mu,\ \sigma,\ \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\ \sigma\ \Gamma\left(\frac{\nu}{2}\right)}\ \left[1+\frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right]^{-\frac{\nu+1}{2}} \tag{2.6}$$

Definition 2.1.6. *A random variable X is said to have a Gaussian distribution with location parameter $\mu \in (-\infty,\ \infty)$ and shape parameter $\theta \in (0, \infty)$ if X has a probability density function of the form*

$$f(x;\ \mu, \theta) = \frac{1}{\sqrt{2\pi\theta}}\ e^{-\frac{1}{2\theta}(x-\mu)^2} \qquad x \in (-\infty, \infty) \tag{2.7}$$

Definition 2.1.7. *A continuous random variable W follows an inverse gamma distribution with shape parameter $\alpha > 0$ and scale parameter $\beta > 0$ if W has a probability density function defined by*

$$f(w;\ \alpha,\ \beta) = \frac{\beta^\alpha}{\Gamma\left(\alpha\right)}\left(\frac{1}{w}\right)^{\alpha+1} e^{-\beta\frac{1}{w}} \qquad w \in (0, \infty) \tag{2.8}$$

Theorem 2.1.8. *Let X be a continuous random variable that follows the Gaussian distribution with location parameter μ and an unknown variance θ, i.e $X \sim N(\mu, \theta)$. Suppose that the variance θ follows an inverse gamma distribution with shape parameter $\alpha = \frac{\nu}{2}$ and scale parameter $\beta = \frac{\nu\sigma^2}{2}$. Then compounding the distribution of X with θ yields a non-standardized t-distribution for the random variable X, with ν degrees of freedom, location parameter μ and a scale parameter σ^2.*

Proof. Assume X and θ are independent. Then the results follow directly from the integration of the joint density function of X and θ over the values of θ. The functions

$$f(x|\theta;\ \mu) = \frac{1}{\sqrt{2\pi\theta}} e^{-\frac{1}{2\theta}(x-\mu)^2} \quad \text{and} \quad f(\theta;\ \nu, \sigma^2) = \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{1}{\theta}\right)^{\frac{\nu}{2}+1} e^{-\frac{\nu\sigma^2}{2\theta}}$$

are the marginal densities of the variables X and θ, respectively. Thus, the joint density function of X and θ is given by $f(x, \theta) = f(x|\theta;\ \mu)f(\theta;\ \nu, \sigma^2)$. Compounding gives

$$\begin{aligned}
f(x;\ \mu, \sigma^2, \nu) &= \int_\theta f(x,\ \theta)\ d\theta \\
&= \int_\theta f(x|\theta;\ \mu) f(\theta;\ \nu, \sigma^2)\ d\theta \\
&= \int_0^\infty \frac{1}{\sqrt{2\pi\theta}} e^{-\frac{1}{2\theta}(x-\mu)^2} \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{1}{\theta}\right)^{\frac{\nu}{2}+1} e^{-\frac{\nu\sigma^2}{2\theta}}\ d\theta \\
&= \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}} \int_0^\infty \frac{1}{\sqrt{\theta}} \left(\frac{1}{\theta}\right)^{\frac{\nu}{2}+1} e^{-\frac{1}{2\theta}\left[\nu\sigma^2+(x-\mu)^2\right]}\ d\theta \\
&= \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}} \int_0^\infty \left(\frac{1}{\theta}\right)^{\frac{\nu}{2}+\frac{1}{2}+1} e^{-\frac{1}{\theta}\frac{1}{2}\left[\nu\sigma^2+(x-\mu)^2\right]}\ d\theta
\end{aligned}$$

Let $\lambda = \frac{1}{\theta}$. Then $d\lambda = -\frac{1}{\theta^2}\ d\theta$ so that $-\left(\frac{1}{\lambda}\right)^2 d\lambda = d\theta$. Substituting and taking note of the respective and corresponding limits of θ and λ, we have

$$\begin{aligned}
f(x;\ \mu, \sigma^2, \nu) &= \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}} \int_0^\infty \left(\frac{1}{\theta}\right)^{\frac{\nu}{2}+\frac{1}{2}+1} e^{-\frac{1}{\theta}\frac{1}{2}\left[\nu\sigma^2+(x-\mu)^2\right]}\ d\theta \\
&= \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}} \int_\infty^0 \lambda^{\frac{\nu}{2}+\frac{1}{2}+1} e^{-\lambda\frac{1}{2}\left[\nu\sigma^2+(x-\mu)^2\right]} \left[-\left(\frac{1}{\lambda}\right)^2\right] d\lambda \\
&= \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}} \int_0^\infty \lambda^{\frac{\nu+1}{2}-1} e^{-\lambda\frac{1}{2}\left[\nu\sigma^2+(x-\mu)^2\right]}\ d\lambda \\
&= \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}} \int_0^\infty \lambda^{\frac{\nu+1}{2}-1} e^{-\lambda\frac{1}{2}\left[\nu\sigma^2+(x-\mu)^2\right]}\ d\lambda \\
&= \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}}\ \mathcal{L}\left(\lambda^{\frac{\nu+1}{2}-1}\right)
\end{aligned}$$

where $\mathcal{L}\left(\lambda^{\frac{\nu+1}{2}-1}\right)$ is the laplace transform of the function of λ given by $g(\lambda) = \left(\lambda^{\frac{\nu+1}{2}-1}\right)$. Thus, we have

$$\begin{aligned} f(x;\ \mu, \sigma^2, \nu) =& \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}}\ \ \mathcal{L}\left(\lambda^{\frac{\nu+1}{2}-1}\right) \\ =& \frac{(\nu\sigma^2)^{\frac{\nu}{2}}}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\sqrt{2\pi}}\ \ \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\left(\frac{1}{2}\right)^{\frac{\nu+1}{2}}\left[\nu\sigma^2 + (x-\mu)^2\right]^{\frac{\nu+1}{2}}} \\ =& \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi\nu\sigma^2}\ \left[1 + \frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right]^{\frac{\nu+1}{2}}} \\ =& \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi\nu\sigma^2}}\ \left[1 + \frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right]^{-\frac{\nu+1}{2}} \end{aligned}$$

which is a probability density function for a non-standardized t-distribution with ν degrees of freedom, location parameter μ and scale parameter σ^2. □

Remark 2.1.2. If a random variable X is such that $X \sim t_1(\mu,\ \sigma^2,\ \nu)$, then

(i) Mean $(X) = \mu$ for $\nu \in (1,\ \infty)$, otherwise $\mathbb{E}(X)$ is undefined .

(ii) Mode $(X) = \mu$

(iii) Median $(X) = \mu$.

(iv) $\mathbb{E}\left[X - \mathbb{E}(X)\right]^2 = \frac{\nu}{\nu-2}\sigma^2$ for $\nu > 2$,

Suppose we have a random vector $\boldsymbol{X}^p = (X_1, \cdots, X_p)^T$ of p random variables such that each univariate variable $X_k \sim t_1(\mu_k,\ \sigma_k^2,\ \nu)$. It can be shown that $\boldsymbol{X}^p$ follows a (multivariate) t-distribution with a $p \times 1$ location parameter vector $\boldsymbol{\mu}$, $p \times p$ scale matrix $\boldsymbol{\Sigma}$ and a scalar degrees of freedom ν.

2.2. The Multivariate t-distribution

The multivariate t-distribution is a multivariate probability distribution which is considered to be a generalization to random vectors of the univariate Student's t-distribution presented in expression (2.6). There are many derivations today for the multivariate generalizations of the Student's t-distribution. A comprehensive study of the multivariate t-distribution, its derivations and other properties is given in [23] and [24]. Here, we recall the definition and review its derivation from the multivariate Gaussian and the univariate Gamma distributions [3].

Definition 2.2.1 ([23]). *A $p-$dimension random vector $\boldsymbol{X}^p = (X_1, \cdots, X_p)^T$ is said to follow a multivariate $t-$distribution with location parameter $\boldsymbol{\mu}$, positive definite scale matrix $\boldsymbol{\Sigma}$ and degrees of freedom ν if it has the probability density function given by*

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\Gamma\left(\frac{\nu+p}{2}\right) |\boldsymbol{\Sigma}|^{-\frac{1}{2}}}{(\pi\nu)^{\frac{p}{2}} \Gamma\left(\frac{\nu}{2}\right)} . \left[1 + \frac{1}{\nu}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right]^{-\frac{\nu+p}{2}} \tag{2.9}$$

If $\boldsymbol{X}^p$ is a p-dimensional random vector that follows a multivariate t-distribution with the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and ν, the notation used is $\boldsymbol{X}^p \sim \mathbf{t}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$.

In this definition, the central location parameter $\boldsymbol{\mu}$ is understood to be a p-dimensional vector, the positive definite scale matrix $\boldsymbol{\Sigma}$ is understood to be a $p \times p$ matrix and the degrees of freedom parameter ν is a positive real scalar from some parameter space $\boldsymbol{\Omega}$. If we let $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)^T$, then the probability density function (2.9), can be written as

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{\Gamma\left(\frac{\nu+p}{2}\right) |\boldsymbol{\Sigma}|^{-\frac{1}{2}}}{(\pi\nu)^{\frac{p}{2}} \Gamma\left(\frac{\nu}{2}\right) \left[1 + \rho(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma})/\nu\right]^{\frac{\nu+p}{2}}} \tag{2.10}$$

where

$$\rho(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \tag{2.11}$$

denotes the Mahalanobis squared distance from the centre $\boldsymbol{\mu}$ to the vector $\boldsymbol{x}$ with respect to the positive definite inner product matrix $\boldsymbol{\Sigma}$. Thus, the multivariate t-distribution is ellipsoidally symmetric about the centre $\boldsymbol{\mu}$, as the density is the same for all $\boldsymbol{x}$ that have the same $\boldsymbol{\Sigma}$ distance from $\boldsymbol{\mu}$ [19].

Definition 2.2.2 ([3]). *A $p-$dimensional vector $\boldsymbol{X}^p = (X_1, \cdots, X_p)^T$ has a multivariate Gaussian distribution with a $p \times 1$ mean vector $\boldsymbol{\mu}$ and a $p \times p$ covariance matrix $\boldsymbol{\Sigma}$ if $\boldsymbol{X}^p$ has the probability density function given by*

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} . \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right\} \tag{2.12}$$

If the random vector $\boldsymbol{X}^p$ follows the multivariate normal distribution, the notation used is $\boldsymbol{X}^p \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Theorem 2.2.3 ([3]). *Let $g(w)$ be a probability density function of the continuous random variable W where $W \sim Gamma\left(\frac{\nu}{2}, \frac{\nu}{2}\right)$ for $0 < w < \infty$. If $p(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the probability density function of a multivariate Gaussian distribution with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, then*

$$\int_0^\infty p(\boldsymbol{x};\ \boldsymbol{\mu}, \boldsymbol{\Sigma}/w)\ d\ g(w; \nu) \tag{2.13}$$

is the probability density function of a multivariate t-distribution with location parameter $\boldsymbol{\mu}$, inner product matrix $\boldsymbol{\Sigma}$ and degrees of freedom ν.

Proof. The standard algebraic operations of integration with respect to w, from the joint density of the variable $\boldsymbol{X}$ and W, lead to the density function of the marginal distribution of $\boldsymbol{X}$. Thus, the results follow directly from the integral

$$I = \int_0^\infty f(\boldsymbol{x};\ \boldsymbol{\mu}, \boldsymbol{\Sigma}/w)\ g(w;\nu)\ dw$$

Thus, we have

$$\begin{aligned}
I =& \int_0^\infty f(\boldsymbol{x};\ \boldsymbol{\mu}, \boldsymbol{\Sigma}/w)\ g(w;\nu)\ dw \\
=& \int_0^\infty \frac{\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{\Sigma}/w)^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}}{(2\pi)^{\frac{p}{2}}\left(\frac{1}{w}\right)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\ \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}} w^{\frac{\nu}{2}-1}e^{-\frac{\nu}{2}w}}{\Gamma\left(\frac{\nu}{2}\right)}\ dw \\
=& \frac{(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \int_0^\infty w^{\frac{\nu+p}{2}-1}e^{-w\left[\frac{\nu}{2}+\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]}\ dw \\
=& \frac{(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \mathcal{L}\left(h(w)\right) \qquad (2.14)
\end{aligned}$$

where $\mathcal{L}\left(h(w)\right)$ is the Laplace transform of the function $h(w)$. In our case, taking $h(w) = w^{\frac{\nu+p}{2}-1}$ and $s = \left[\frac{\nu}{2} + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]$, the integral $\int_0^\infty w^{\frac{\nu+p}{2}-1}e^{-ws}\ dw$ can be evaluated using the Laplace transform of the function $h(w) = w^{\frac{\nu+p}{2}-1}$, so that

$$\begin{aligned}
I =& \frac{(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \mathcal{L}\left[w^{\frac{\nu+p}{2}-1}\right] \\
=& \frac{(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \frac{\Gamma\left(\frac{\nu+p}{2}\right)}{s^{\frac{\nu+p}{2}}} \\
=& \frac{(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \frac{\Gamma\left(\frac{\nu+p}{2}\right)}{\left[\frac{\nu}{2} + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]^{\frac{\nu+p}{2}}} \\
=& \frac{(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}\Gamma\left(\frac{\nu+p}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\left(\frac{\nu}{2}\right)^{\frac{\nu+p}{2}}\left[1 + \frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]^{\frac{\nu+p}{2}}} \\
=& \frac{|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}\Gamma\left(\frac{\nu+p}{2}\right)}{(2\pi)^{\frac{p}{2}}\Gamma\left(\frac{\nu}{2}\right)\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}\left(\frac{\nu}{2}\right)^{\frac{p}{2}}\left[1 + \frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]^{\frac{\nu+p}{2}}} \\
=& \frac{|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\Gamma\left(\frac{\nu+p}{2}\right)}{(2\pi)^{\frac{p}{2}}\left(\frac{\nu}{2}\right)^{\frac{p}{2}}\Gamma\left(\frac{\nu}{2}\right)\left[1 + \frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]^{\frac{\nu+p}{2}}} \\
=& \frac{|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\Gamma\left(\frac{\nu+p}{2}\right)}{(\nu\pi)^{\frac{p}{2}}\Gamma\left(\frac{\nu}{2}\right)\left[1 + \frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]^{\frac{\nu+p}{2}}} \\
=& f(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\nu)
\end{aligned}$$

Thus,

$$\int_0^\infty p(\boldsymbol{x};\ \boldsymbol{\mu}, \boldsymbol{\Sigma}/w)\ d\ g(w;\nu) = f(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\nu) \tag{2.15}$$

which is the probability density function of a multivariate t-distribution with location parameter $\boldsymbol{\mu}$, positive definite inner product matrix $\boldsymbol{\Sigma}$ and degrees of freedom ν. □

Remark 2.2.1 ([19])**.** If $\boldsymbol{X}^p$ is a random vector such that $\boldsymbol{X}^p \sim \mathbf{t}_p(\boldsymbol{\mu},\ \boldsymbol{\Sigma},\ \boldsymbol{\nu})$, then the following properties hold;

(i) $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{\mu}$ provided $\nu > 1$, otherwise $\mathbb{E}[\boldsymbol{X}]$ is undefined

(ii) Mode $[\boldsymbol{X}] = \boldsymbol{\mu}$

(iii) $\mathbb{E}[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T] = \left(\frac{\nu}{\nu-2}\right)\boldsymbol{\Sigma}$ provided $\nu > 2$.

(iv) As $\nu \to \infty$, then $W \to 1$ with probability 1, and $\boldsymbol{X}^p$ becomes marginally Normal with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

(v) Further,

$$\boldsymbol{X}|\ \boldsymbol{\mu},\boldsymbol{\Sigma},\nu,w\ \sim N_p\left(\boldsymbol{\mu},\ \boldsymbol{\Sigma}/w\right) \tag{2.16}$$

and

$$W|\ \boldsymbol{\mu},\boldsymbol{\Sigma},\nu\ \sim Gamma\left(\frac{\nu}{2},\ \frac{\nu}{2},\right) \tag{2.17}$$

Thus,

$$f\left(\boldsymbol{x}|\ \boldsymbol{\mu},\boldsymbol{\Sigma},\nu,w\right) = \frac{(w)^{\frac{p}{2}}}{(2\pi)^{\frac{p}{2}}\left|\boldsymbol{\Sigma}\right|^{\frac{1}{2}}}.\exp\left\{-\frac{w}{2}\rho(\boldsymbol{x},\boldsymbol{\mu};\boldsymbol{\Sigma})\right\} \tag{2.18}$$

and

$$f(w|\boldsymbol{\mu},\boldsymbol{\Sigma},\nu) = \frac{\frac{\nu}{2}^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} w^{\frac{\nu}{2}-1}e^{-\frac{\nu}{2}w} \tag{2.19}$$

where $\rho(\boldsymbol{x},\boldsymbol{\mu};\boldsymbol{\Sigma})$ is as defined in (2.11).

2.3. Mixtures of Multivariate t-distributions

Definition 2.3.1 ([3])**.** *Let $\left\{\boldsymbol{X}_j^p \mid \text{ for } j = 1,\cdots,g\right\}$ denote a finite class of p-dimensional random vectors such that $\boldsymbol{X}_j^p \sim \mathbf{t}_p\left(\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j,\nu_j\right)$ for some $\boldsymbol{\theta}_j = \left(\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j,\nu_j\right)^T \in \boldsymbol{\Omega}$. Further, let $f(\boldsymbol{x}_j;\ \boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j,\nu_j)$ denote the density function of $\boldsymbol{X}_j^p$. Then a p-dimensional random vector $\boldsymbol{X}^p$ follows a $g-$component mixture distribution of multivariate t-distributions if $\boldsymbol{X}^p$ has density function:*

$$\begin{aligned}\mathcal{F}(\boldsymbol{x};\ \boldsymbol{\Psi}) &= \sum_{j=1}^{g}\tau_j f(\boldsymbol{x};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j,\nu_j)\\ &= \sum_{j=1}^{g}\frac{\tau_j\Gamma\left(\frac{\nu_j+p}{2}\right)\left|\boldsymbol{\Sigma}_j\right|^{-\frac{1}{2}}}{(\pi\nu_j)^{\frac{p}{2}}\Gamma\left(\frac{\nu_j}{2}\right)\left[1+\rho(\boldsymbol{x},\ \boldsymbol{\mu}_j;\ \boldsymbol{\Sigma}_j)/\nu_j\right]^{\frac{\nu_j+p}{2}}}\end{aligned} \tag{2.20}$$

where $\tau_j > 0$ *is the mixing proportion for the* j^{th} *component such that* $\sum_{j=1}^{g} \tau_j = 1$, $\boldsymbol{\Psi} = \left(\tau_1, \cdots, \tau_{g-1}, \boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_g^T\right)^T$ *is the vector containing all the model parameters and* $\boldsymbol{\theta}_j^T = \left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \nu_j\right)$ *contains the parameters for the* j^{th} *component of the mixture model, for* $j = 1, \cdots, g$.

Note 2.3.1. From the definition of the mixture model represented by the mixture density function (2.20), we emphasize the following:

(i) $f(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \nu_j) = f(\boldsymbol{x}_i;\ \boldsymbol{\theta}_j)$ is the density for the j^{th} component.

(ii) $\boldsymbol{\theta}_j$ contains the elements of the location parameter (mean) $\boldsymbol{\mu}_j$, distinct elements of the positive definite inner product matrix $\boldsymbol{\Sigma}_j$ and the degrees of freedom parameter ν_j, for the j^{th} component

(iii) $\boldsymbol{X}^p$ is a random p-dimensional vector and $\boldsymbol{x}^p$ is its specific observation which we simply write as $\boldsymbol{X}$ and $\boldsymbol{x}$ respectively, as the dimension p is understood.

(iv) In the above sequel, we are using f as a generic symbol for a probability density function (p.d.f.)

(v) $\rho(\boldsymbol{x},\ \boldsymbol{\mu}_j;\ \boldsymbol{\Sigma}_j)$ is the Mahalanobis distance from $\boldsymbol{x}$ to the center $\boldsymbol{\mu}_j$ with respect to $\boldsymbol{\Sigma}_j$

2.4. Maximum Likelihood Estimation of Parameters in Mixtures of Multivariate t-distributions

In maximum likelihood (ML) estimation, we wish to find an estimate $\hat{\boldsymbol{\Psi}}$ of the parameter $\boldsymbol{\Psi}$, which maximizes the likelihood of the observed data. This section discusses the ML estimation of parameters in mixtures of multivariate t-distributions.

Let $\boldsymbol{X}$ be a p-dimensional mixture random vector with probability density function (2.20). Further, let $\boldsymbol{X}_o = (\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n)^T$ denote a compact representation of p-dimensional independent random sample of size n and let $\boldsymbol{x_o} = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$ denote an observed $p-$dimensional independent sample of size n from a g-component finite mixture distribution with probability density function as defined in (2.20). The data point $\boldsymbol{x}_i$ for $i = 1, 2, \cdots, n$, is a realization of the random mixture vector $\boldsymbol{X}_i$. Let $\mathcal{F}(\boldsymbol{x}|\boldsymbol{\Psi})$ denote the g-component mixture probability density function of $\boldsymbol{X}$ as defined in (2.20), where $\boldsymbol{\Psi}$ is a vector containing all the unknown parameters of the mixture model.

Definition 2.4.1. *Let* $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$ *be a random sample of size n from the mixture distribution (2.20). The likelihood function, denoted* $L(\boldsymbol{\Psi})$,

is the joint probability density function of $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$*, which is considered to be a function of the mixture model parameter* $\boldsymbol{\Psi}$ *and is defined by*

$$\begin{aligned} L(\boldsymbol{\Psi}) &= \prod_{i=1}^{n} \mathcal{F}(\boldsymbol{x}_i|\boldsymbol{\Psi}) \\ &= \prod_{i=1}^{n} \sum_{j=1}^{g} \tau_j f(\boldsymbol{x}_i|\boldsymbol{\theta}_j) \end{aligned} \tag{2.21}$$

Definition 2.4.2. *Let* $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$ *be a random sample of size n from the mixture distribution (2.20) with likelihood function* $L(\boldsymbol{\Psi})$*. The logarithm of the likelihood function, denoted* $l(\boldsymbol{\Psi})$*, is called the log-likelihood function*

$$\begin{aligned} l(\boldsymbol{\Psi}) &= \ln[L(\boldsymbol{\Psi})] \\ &= \ln \prod_{i=1}^{n} \sum_{j=1}^{g} \tau_j f(\boldsymbol{x}_i|\boldsymbol{\theta}_j) \\ &= \sum_{i=1}^{n} \ln \sum_{j=1}^{g} \tau_j f(\boldsymbol{x}_i|\boldsymbol{\theta}_j) \end{aligned} \tag{2.22}$$

Given the observed data set $\boldsymbol{x_o} = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$, the likelihood function $L(\boldsymbol{\Psi})$ can be written as $L(\boldsymbol{\Psi};\ \boldsymbol{x_o})$. Similarly, we can write the corresponding log-likelihood function as $l(\boldsymbol{\Psi};\ \boldsymbol{x_o})$. The distribution of the components in (2.21) and (2.22) are t-distributions so that the mixture is given by

$$\mathcal{F}(\boldsymbol{x}_i|\boldsymbol{\Psi}) = \sum_{j=1}^{g} \frac{\tau_j \Gamma\left(\frac{\nu_j+p}{2}\right) |\boldsymbol{\Sigma}_j|^{-\frac{1}{2}}}{(\pi\nu_j)^{\frac{p}{2}} \Gamma\left(\frac{\nu_j}{2}\right) \left[1 + \rho(\boldsymbol{x}_i,\ \boldsymbol{\mu}_j;\ \boldsymbol{\Sigma}_j)/\nu_j\right]^{\frac{\nu_j+p}{2}}}$$

Thus we have

$$L(\boldsymbol{\Psi}) = \prod_{i=1}^{n} \sum_{j=1}^{g} \frac{\tau_j \Gamma\left(\frac{\nu_j+p}{2}\right) |\boldsymbol{\Sigma}_j|^{-\frac{1}{2}}}{(\pi\nu_j)^{\frac{p}{2}} \Gamma\left(\frac{\nu_j}{2}\right) \left[1 + \rho(\boldsymbol{x}_i,\ \boldsymbol{\mu}_j;\ \boldsymbol{\Sigma}_j)/\nu_j\right]^{\frac{\nu_j+p}{2}}} \tag{2.23}$$

and

$$l(\boldsymbol{\Psi}) = \sum_{i=1}^{n} \ln \sum_{j=1}^{g} \frac{\tau_j \Gamma\left(\frac{\nu_j+p}{2}\right) |\boldsymbol{\Sigma}_j|^{-\frac{1}{2}}}{(\pi\nu_j)^{\frac{p}{2}} \Gamma\left(\frac{\nu_j}{2}\right) \left[1 + \rho(\boldsymbol{x}_i,\ \boldsymbol{\mu}_j;\ \boldsymbol{\Sigma}_j)/\nu_j\right]^{\frac{\nu_j+p}{2}}} \tag{2.24}$$

where $\boldsymbol{\Psi} = \left(\tau_1, \cdots, \tau_{g-1}, \boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_g^T\right)^T$, $0 < \tau_j < 1$, $\sum_{j=1}^{g} \tau_j = 1$, and $\boldsymbol{\theta}_j = \left(\boldsymbol{\mu}_j \boldsymbol{\Sigma}_j, \nu_j\right)^T$ for $j = 1, \cdots, g$.

Definition 2.4.3. *Let* $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$ *be a random sample of size n from the mixture distribution (2.20) with likelihood function* $L(\boldsymbol{\Psi})$ *and log-likelihood function* $l(\boldsymbol{\Psi})$*. The score function, denoted by* $S(\boldsymbol{\Psi})$*, is the first order derivative of the log-likelihood function*

$$S(\boldsymbol{\Psi}) = \frac{\partial}{\partial \boldsymbol{\Psi}} l(\boldsymbol{\Psi}; \boldsymbol{x}_o) = \frac{1}{L(\boldsymbol{\Psi}; \boldsymbol{x}_o)} \frac{\partial L(\boldsymbol{\Psi}; \boldsymbol{x}_o)}{\partial \boldsymbol{\Psi}} \tag{2.25}$$

The maximum likelihood estimate (MLE) of $\boldsymbol{\Psi}$ is obtained by solving:

$$\frac{1}{L(\boldsymbol{\Psi}; \boldsymbol{x}_o)} \frac{\partial L(\boldsymbol{\Psi}; \boldsymbol{x}_o)}{\partial \boldsymbol{\Psi}} = 0 \tag{2.26}$$

We denote the solution to equation (2.26) by

$$\hat{\boldsymbol{\Psi}} = \left(\hat{\boldsymbol{\theta}}^T, \ \hat{\boldsymbol{\tau}}^T \right)^T \tag{2.27}$$

where

$$\hat{\boldsymbol{\theta}} = \left(\hat{\boldsymbol{\theta}}_1^T, \cdots, \hat{\boldsymbol{\theta}}_g^T \right)^T \qquad \text{and} \qquad \hat{\boldsymbol{\tau}} = (\hat{\tau}_1, \cdots, \hat{\tau}_{g-1})^T$$

Maximization of the observed log-likelihood function (2.22) to obtain the maximum likelihood estimates (2.27), is difficult [1][4]; For example, to determine $\hat{\boldsymbol{\theta}}_j$, the MLE for the parameter of the j^{th} component, we have

$$\begin{aligned}
\frac{\partial l(\boldsymbol{\Psi}; \ \boldsymbol{x}_o)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \ln \sum_{k=1}^{g} \tau_k f(\boldsymbol{x}_i | \boldsymbol{\theta}_k) \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} \ln \sum_{k=1}^{g} \tau_k f(\boldsymbol{x}_i | \boldsymbol{\theta}_k) \\
&= \sum_{i=1}^{n} \frac{1}{\sum_{k=1}^{g} \tau_k f(\boldsymbol{x}_i | \boldsymbol{\theta}_k)} \ \tau_j \ \frac{\partial}{\partial \theta_j} f(\boldsymbol{x}_i | \boldsymbol{\theta}_j) \\
&= \sum_{i=1}^{n} \frac{\tau_j f(\boldsymbol{x}_i | \boldsymbol{\theta}_j)}{\sum_{k=1}^{g} \tau_k f(\boldsymbol{x}_i | \boldsymbol{\theta}_k)} \ \frac{1}{f(\boldsymbol{x}_i | \boldsymbol{\theta}_j)} \ \frac{\partial}{\partial \theta_j} f(\boldsymbol{x}_i | \boldsymbol{\theta}_j) \\
&= \sum_{i=1}^{n} \frac{\tau_j f(\boldsymbol{x}_i | \boldsymbol{\theta}_j)}{\sum_{k=1}^{g} \tau_k f(\boldsymbol{x}_i | \boldsymbol{\theta}_k)} \ \frac{\partial}{\partial \theta_j} \ln f(\boldsymbol{x}_i | \boldsymbol{\theta}_j)
\end{aligned} \tag{2.28}$$

Note that for a single component model with parameter say $\boldsymbol{\theta}_j$, the derivative of the log-likelihood function is just given by

$$\sum_{i=1}^{n} \ \frac{\partial}{\partial \theta_j} \ln f(\boldsymbol{x}_i | \boldsymbol{\theta}_j) \tag{2.29}$$

From (2.28), we see that maximizing the log-likelihood for a g-component mixture model is like performing a weighted likelihood maximization, where the weights of $\boldsymbol{x}_i$ are given by

$$e_{ij} = \frac{\tau_j f(\boldsymbol{x}_i | \boldsymbol{\theta}_j)}{\sum_{j=k}^{g} \tau_k f(\boldsymbol{x}_i | \boldsymbol{\theta}_k)} \tag{2.30}$$

which unfortunately, depend on the parameters that we wish to estimate. Thus, closed form MLEs are not readily available [4]. This problem is best dealt with by assuming that there are some missing variables which should specify the component of origin of the observed data set $\boldsymbol{x}_i$ [1]. The introduction of missing data and application of an iterative procedure such as the Newton-Raphson or the EM algorithm to compute the parameter estimates provides helpful estimation procedures in such situations [1][2]. The proceeding chapter discusses the EM algorithm and its application in computing the MLE estimates from equation (2.28).

CHAPTER 3
The Expectation Maximization Algorithm

This Chapter reviews the Expectation Maximization (EM) algorithm and its application in the computation of ML estimates of parameters in mixture models. Section 3.1 reviews the general form of the EM algorithm and its underlying theory [1][2][35]. Section 3.2 reviews the details of the EM algorithm as used in parameter estimation for mixtures of multivariate t-distributions. The details of the E-step and the M-step of the EM process are thoroughly presented. Section 3.3 reviews some convergence properties of the EM algorithm.

3.1. General EM Algorithm

The Generalised Expectation Maximization (GEM) algorithm was officially described and presented by Dempster, Laird and Rubin (1977) [2]. In their paper, Dempster et al (1977) describe the expectation step (E-step) and the maximization step (M-step) in their general forms, give some theoretical properties of the EM algorithm and identify and give a wide range of applications of the EM algorithm in statistics. We present a review of the GEM algorithm as presented in [2]. For mathematically detailed discussion on the contents of this Section, the reader is referred to [1][2] and [13].

Let $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{M}}$ denote two sample spaces such that the incomplete data $\boldsymbol{X}_o$ has its possible realizations from $\boldsymbol{\mathcal{X}}$ and the corresponding complete data $\boldsymbol{X}_c$ has realizations from $\boldsymbol{\mathcal{M}}$. Further, postulate a many-to-one mapping from $\boldsymbol{\mathcal{M}}$ to $\boldsymbol{\mathcal{X}}$ given by

$$\boldsymbol{X}_c \rightarrow \boldsymbol{X}_o(\boldsymbol{X}_c) \tag{3.1}$$

For each observed incomplete data set $\boldsymbol{x}_o \in \boldsymbol{\mathcal{X}}$, the corresponding complete data set $\boldsymbol{x}_c$ is known only to lie in $\boldsymbol{\mathcal{M}}(\boldsymbol{x}_o) \subseteq \boldsymbol{\mathcal{M}}$ where

$$\boldsymbol{\mathcal{M}}(\boldsymbol{x}_o) = \{\boldsymbol{x}_c : \ \boldsymbol{x}_o(\boldsymbol{x}_c) = \boldsymbol{x}_o\} \tag{3.2}$$

The complete data set $\boldsymbol{x}_c$ is not observed directly, but indirectly through the incomplete data set $\boldsymbol{x}_o$ from $\boldsymbol{\mathcal{X}}$. Further, let $f(\boldsymbol{x}_c;\ \boldsymbol{\Psi})$ denote the probability density function associated with the complete data random variable $\boldsymbol{X}_c$ for

$\boldsymbol{\Psi} \in \boldsymbol{\Omega}$ where $\boldsymbol{\Psi}$ is the parameter space. Then the probability density function of the random variable $\boldsymbol{X}_o$ associated with incomplete data set $\boldsymbol{x}_o$ denoted by $g(\boldsymbol{x}_o;\ \boldsymbol{\Psi})$ is given by

$$g(\boldsymbol{x}_o;\ \boldsymbol{\Psi}) = \int_{\boldsymbol{\mathcal{M}}(\boldsymbol{x}_o)} f(\boldsymbol{x}_c;\ \boldsymbol{\Psi})\ d\boldsymbol{x}_c \tag{3.3}$$

The EM algorithm aims to find the value of $\boldsymbol{\Psi}$ in $\boldsymbol{\Omega}$ that maximizes the function $g(\boldsymbol{x}_o;\ \boldsymbol{\Psi})$ given the observed data set $\boldsymbol{x}_o$. Maximizing $g(\boldsymbol{x}_o;\ \boldsymbol{\Psi})$ directly is often a difficulty task for many statistical problems [2][1]. Interestingly, maximizing the pdf $f(\boldsymbol{x}_c;\ \boldsymbol{\Psi})$ associated with the complete data is simpler than maximizing the pdf $g(\boldsymbol{x}_o;\ \boldsymbol{\Psi})$ associated with the observed data set which is the incomplete data set [13]. Thus, the EM algorithm maximizes $g(\boldsymbol{x}_o; \boldsymbol{\Psi})$ indirectly through maximizing $f(\boldsymbol{x}_c;\ \boldsymbol{\Psi})$ directly. This requires that we obtain the complete data log-likelihood function, $\log f(\boldsymbol{x}_c;\ \boldsymbol{\Psi})$. However, since the complete data $\boldsymbol{x}_c$ is not observed, the log-likelihood of its specification $\log f(\boldsymbol{x}_c; \boldsymbol{\Psi})$ is replaced by its conditional expectation given $\boldsymbol{x}_o$ and the current value of $\boldsymbol{\Psi}$. Thus, for all pairs $(\boldsymbol{\Psi}',\ \boldsymbol{\Psi})$, define the function

$$Q(\boldsymbol{\Psi}'|\boldsymbol{\Psi}) = E\left[\log f(\boldsymbol{x}_c|\boldsymbol{\Psi}')|\ \boldsymbol{x}_o, \boldsymbol{\Psi}\right] \tag{3.4}$$

Further, let $k(\boldsymbol{x}_c|\boldsymbol{x}_o, \boldsymbol{\Psi}) = f(\boldsymbol{x}_c;\ \boldsymbol{\Psi})/g(\boldsymbol{x}_o;\ \boldsymbol{\Psi})$ denote the conditional density of $\boldsymbol{X}_c$ given $\boldsymbol{X}_o = \boldsymbol{x}_o$ and $\boldsymbol{\Psi}$ so that for the pairs $(\boldsymbol{\Psi}',\ \boldsymbol{\Psi})$, define the function $H(\boldsymbol{\Psi}'|\boldsymbol{\Psi})$ as

$$H(\boldsymbol{\Psi}'|\boldsymbol{\Psi}) = E\left[\log k(\boldsymbol{x}_c|\ \boldsymbol{x}_o, \boldsymbol{\Psi}')|\ \boldsymbol{x}_o, \boldsymbol{\Psi}\right] \tag{3.5}$$

It can easily be seen that the log-likelihood function denoted $l(.)$ can be written as

$$l(\boldsymbol{\Psi}') = \log g(\boldsymbol{x}_o|\boldsymbol{\Psi}') = Q(\boldsymbol{\Psi}'|\boldsymbol{\Psi}) - H(\boldsymbol{\Psi}'|\boldsymbol{\Psi}) \tag{3.6}$$

The EM algorithm iteration $\boldsymbol{\Psi}^{(m)} \rightarrow \boldsymbol{\Psi}^{(m+1)} \in \boldsymbol{M}(\boldsymbol{\Psi}^{(m)})$ can be defined as follows:

(i) E-step: Determine $Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)})$, the current conditional expectation of the complete data log-likelihood after the m^{th} iteration.

(ii) M-step: Choose $\boldsymbol{\Psi}^{(m+1)}$ to be any value of $\boldsymbol{\Psi} \in \boldsymbol{\Omega}$ which maximizes $Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)})$.

where $\boldsymbol{M}(\boldsymbol{\Psi}^{(m)})$ is the set of values which maximizes $Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)})$ over the parameter space $\boldsymbol{\Omega}$. At times, performing the M-step as described above may not be numerically feasible [13]. In view of this, Dempster et al (1977) defined a more generalized EM algorithm (a GEM algorithm) to be an iterative scheme $\boldsymbol{\Psi}^{(m)} \rightarrow \boldsymbol{\Psi}^{(m+1)} \in \boldsymbol{M}(\boldsymbol{\Psi}^{(m)})$ where $\boldsymbol{\Psi} \rightarrow \boldsymbol{M}(\boldsymbol{\Psi})$ is a point-to-set map, such that

$$Q(\boldsymbol{\Psi}'|\boldsymbol{\Psi}) \geq Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}) \qquad \forall\ \boldsymbol{\Psi}' \in \boldsymbol{M}(\boldsymbol{\Psi}^{(m)}) \tag{3.7}$$

and consequently

$$H(\boldsymbol{\Psi}|\boldsymbol{\Psi}) \geq H(\boldsymbol{\Psi}'|\boldsymbol{\Psi}) \qquad \forall\ \boldsymbol{\Psi}' \in \boldsymbol{\Omega} \tag{3.8}$$

From (3.7) and (3.8), it is easy to see that the GEM algorithm generates a sequence of parameters $\{\boldsymbol{\Psi}^{(m)}\}_{m=0}$ such that

$$l(\boldsymbol{\Psi}^{(m+1)}) \geq l(\boldsymbol{\Psi}^{(m)}) \tag{3.9}$$

In finite mixtures, the observed data set $\boldsymbol{x}_o = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$ of size n is considered to be a realization from the sample space $\boldsymbol{\mathcal{X}}$. We assume that there exists g finite states and each observed value $\boldsymbol{x}_i$ for $i = 1, \cdots, n$ is assumed to be associated with only one of the g finite states, which is unknown. This makes $\boldsymbol{x}_o$ to be an "incomplete" data set. The missing data $\boldsymbol{z} = \left(\boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T\right)^T$ is introduced where $\boldsymbol{z}_i$ is a g-dimensional vector whose entries are all zeros except for one equal to unity indicating the unobserved state associated with $\boldsymbol{x}_i$. The complete data is now specified as $\boldsymbol{x}_c = \left(\boldsymbol{x}_o^T, \boldsymbol{z}^T\right)^T$ and is considered to be a realization from the sample space $\boldsymbol{\mathcal{M}}$. With this specification of the complete data, the EM algorithm can be implemented for a g-component finite mixture model.

3.2. Application to Mixtures of Multivariate t-distributions

The use of the EM algorithm in maximum likelihood estimation of parameters for multivariate t-distributions has been considered in a probability model featuring only one t-distributed component [19] [8]. The employed methods showed that the variants of the EM algorithm called the Expectation Conditional Maximization (ECM) and the Expectation Conditional Maximization Either (ECME) algorithms converge faster than the General EM algorithm, thus providing better estimation procedures. A detailed documentation of maximum likelihood estimation of parameters for single component models of t-distribution is given in [1],[8] and [19]. [3] provides a thorough extension of the methods of [1],[8] and [19] to the general g-component mixtures of multivariate t-distributions. In their methods, [3] employ the ECM algorithm when estimating the parameter for the degrees of freedom $\boldsymbol{\nu}$.

We now review the application of the EM (and ECM) algorithm in the estimation of model parameters for a g-component mixture of multivariate t-distributions as presented in [3].

Let the multivariate t-distribution mixture model be given by:

$$\mathcal{F}(\boldsymbol{x}_i|\boldsymbol{\Psi}) = \sum_{j=1}^{g} \tau_j f(\boldsymbol{x}_i|\boldsymbol{\theta}_j) \tag{3.10}$$

where $\boldsymbol{\Psi} = (\tau_1, \cdots, \tau_{g-1}, \boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_g^T)^T$ and $\boldsymbol{\theta}_j$ is the parameter for the j^{th} component. In this case, $\boldsymbol{\theta}_j$ contains the elements of $\boldsymbol{\mu}_j$, distinct elements of $\boldsymbol{\Sigma}_j$ and the degrees of freedom parameter ν_j i.e $\boldsymbol{\theta}_j = \left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \nu_j\right)^T$ for $j = 1, \cdots, g$. For computational convenience, we represent $\boldsymbol{\Psi}$ as:

$$\boldsymbol{\Psi} = \left(\boldsymbol{\tau}^T, \boldsymbol{\theta}^T\right)^T = \left(\boldsymbol{\tau}^T, \boldsymbol{\lambda}^T, \boldsymbol{\nu}^T\right)^T \tag{3.11}$$

where $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_g)^T$ is the vector of component membership probabilities, $\boldsymbol{\nu} = (\nu_1, \cdots, \nu_g)^T$ contains the degrees of freedom for each group and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1^T, \cdots, \boldsymbol{\lambda}_g^T)^T$ where $\boldsymbol{\lambda}_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)^T$ contains the distinct elements of the location parameter $\boldsymbol{\mu}_j$ and the distinct elements of the correlation matrix $\boldsymbol{\Sigma}_j$. Note that $\boldsymbol{\theta} = \left(\boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_g^T\right)^T = \left(\boldsymbol{\lambda}^T, \boldsymbol{\nu}^T\right)^T$.

The observed data set $\boldsymbol{x}_o = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$ is considered to be an incomplete data set. The corresponding likelihood function (2.21), is called the incomplete data likelihood function. It can be seen from (2.28) that MLE estimation of $\boldsymbol{\Psi}$ is not tractable when we try to maximize the incomplete data log-likelihood (2.22). To make the ML estimation of $\boldsymbol{\Psi}$ tractable, the missing data (latent data) $\boldsymbol{Z} = \left(\boldsymbol{Z}_1^T, \cdots, \boldsymbol{Z}_n^T\right)^T$ are introduced into the model where the g-dimensional random vector $\boldsymbol{Z}_i = (Z_{i1}, \cdots, Z_{ig})^T$ is an indicator variable which indicates the component of origin of data point $\boldsymbol{x}_i$ where

$$Z_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ belongs to the } j^{th} \text{ component} \\ \\ 0 & \text{otherwise .} \end{cases} \tag{3.12}$$

The $n \times g$ random matrix $\boldsymbol{Z}$ is a compact representation of the random variables Z_{ij}. The lowercase letters $\boldsymbol{z}$, $\boldsymbol{z}_i$ and z_{ij}, will denote a specific realization of the variables $\boldsymbol{Z}$, $\boldsymbol{Z}_i$ and Z_{ij}, respectively. In the EM-framework, given the observed data set $\boldsymbol{x}_o$, we introduce the latent vector $\boldsymbol{z} = \left(\boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T\right)^T$. The 'partially complete' data vector denoted by $\boldsymbol{x}_{pc}$, is now given as:

$$\boldsymbol{x}_{pc} = \left(\boldsymbol{x}_o^T,\ \boldsymbol{z}^T\right)^T = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T,\ \boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T\right)^T \tag{3.13}$$

Recall that τ_j is the group membership probability for the j^{th} component. From (3.12), we see that for any observed data point $\boldsymbol{x}_i \quad \forall i = 1, \cdots, n$,

$$Pr\left(Z_{ij} = 1 |\ \boldsymbol{\Psi}\right) = \tau_j \tag{3.14}$$

Any specific realization or value of the random vector $\boldsymbol{Z}_i = (Z_{i1}, \cdots, Z_{ig})^T$ say $\boldsymbol{z}_i = (z_{i1}, \cdots, z_{ig})^T$, will have g entries that are all zeros except for one entry which will be a one. Thus, given the group membership parameter $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_g)^T$, it can easily be seen that the conditional density of $\boldsymbol{Z}_i$ is a Multinomial density, i.e $\boldsymbol{Z}_i \sim$ Mult $(1, \tau_1, \cdots, \tau_g)$ (see [4]) so that we have;

$$\begin{aligned} f(\boldsymbol{z}_i|\boldsymbol{\tau}) =& \frac{1!}{z_{i1}!.z_{i2}! \ldots z_{i(g-1)}!.z_{ig}!} \tau_1^{z_{i1}} \tau_2^{z_{i2}} \ldots \tau_g^{z_{ig}} \\ =& \tau_1^{z_{i1}}.\tau_2^{z_{i2}} \ldots \tau_{(g-1)}^{z_{i(g-1)}}.\tau_g^{z_{ig}} \\ =& \prod_{j=1}^{g} \tau_j^{z_{ij}} \end{aligned} \tag{3.15}$$

Further, given the vector $\boldsymbol{z}_i$, all entries z_{ij} are zeros except one which will have the value one. Hence, the conditional density of $\boldsymbol{X}_i$ given $\boldsymbol{Z}_i = \boldsymbol{z}_i$ can

be written as

$$f(\boldsymbol{x}_i|\boldsymbol{z}_i,\boldsymbol{\theta}) = \prod_{j=1}^{g} [f(\boldsymbol{x}_i|\boldsymbol{\theta}_j)]^{z_{ij}} \tag{3.16}$$

so that the joint probability density function of $\boldsymbol{X}_i$ and $\boldsymbol{Z}_i$ is written as

$$\begin{aligned} f(\boldsymbol{x}_i,\boldsymbol{z}_i|\boldsymbol{\theta},\boldsymbol{\tau}) =& f(\boldsymbol{x}_i|\boldsymbol{z}_i,\boldsymbol{\theta}) f(\boldsymbol{z}_i|\boldsymbol{\tau}) \\ =& \prod_{j=1}^{g} [f(\boldsymbol{x}_i|\boldsymbol{z}_i,\boldsymbol{\theta}_j)]^{z_{ij}} \prod_{j=1}^{g} \tau_j^{z_{ij}} \\ =& \prod_{j=1}^{g} [\tau_j f(\boldsymbol{x}_i|\boldsymbol{z}_i,\boldsymbol{\theta}_j)]^{z_{ij}} \end{aligned}$$

Hence, the joint density of the random variables $\boldsymbol{X}_i$ and the latent variable $\boldsymbol{Z}_i$ given the mixture model parameter vector $\boldsymbol{\Psi}$ is

$$f(\boldsymbol{x}_i,\boldsymbol{z}_i|\boldsymbol{\Psi}) \quad = \quad \prod_{j=1}^{g} [\tau_j f(\boldsymbol{x}_i|\boldsymbol{z}_i,\boldsymbol{\theta}_j)]^{z_{ij}} \tag{3.17}$$

Using the joint p.d.f (3.17), we can generate the likelihood function of the partially complete random sample $\boldsymbol{X}_{pc} = \left(\boldsymbol{X}_o^T,\ \boldsymbol{Z}^T\right)^T$, denoted $L_{pc}(\boldsymbol{\Psi})$, using the 'partially complete' data vector (3.13):

$$\begin{aligned} L_{pc}(\boldsymbol{\Psi}) =& \prod_{i=1}^{n} \prod_{j=1}^{g} [\tau_j f(\boldsymbol{x}_i|\boldsymbol{\theta}_j)]^{z_{ij}} \\ =& \prod_{i=1}^{n} \prod_{j=1}^{g} \left[\frac{\tau_j \Gamma\left(\frac{\nu_j+p}{2}\right) |\boldsymbol{\Sigma}_j|^{-\frac{1}{2}}}{(\pi\nu_j)^{\frac{p}{2}} \Gamma\left(\frac{\nu_j}{2}\right) \left[1+\rho(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)/\nu_j\right]^{\frac{\nu_j+p}{2}}} \right]^{z_{ij}} \end{aligned} \tag{3.18}$$

The likelihood function (3.18) may also be denoted by $L_{pc}(\boldsymbol{\Psi}) = L_{pc}(\boldsymbol{\Psi};\ \boldsymbol{x_{pc}})$. The corresponding log-likelihood function, denoted $l_{pc}(\boldsymbol{\Psi}) = l_{pc}(\boldsymbol{\Psi};\ \boldsymbol{x_{pc}})$, is determined by taking the logarithm of the function (3.18). Thus we have:

$$\begin{aligned} l_{pc}(\boldsymbol{\Psi}) =& \ln \prod_{i=1}^{n} \prod_{j=1}^{g} [\tau_j f(\boldsymbol{x}_i|\boldsymbol{\theta}_j)]^{z_{ij}} \\ =& \ln \prod_{i=1}^{n} \prod_{j=1}^{g} \left[\frac{\tau_j \Gamma\left(\frac{\nu_j+p}{2}\right) |\boldsymbol{\Sigma}_j|^{-\frac{1}{2}}}{(\pi\nu_j)^{\frac{p}{2}} \Gamma\left(\frac{\nu_j}{2}\right) \left[1+\rho(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)/\nu_j\right]^{\frac{\nu_j+p}{2}}} \right]^{z_{ij}} \\ =& \sum_{i=1}^{n} \sum_{j=1}^{g} z_{ij} \left[\ln \tau_j + \ln \frac{(\pi\nu_j)^{-\frac{p}{2}} \Gamma\left(\frac{\nu_j+p}{2}\right) |\boldsymbol{\Sigma}_j|^{-\frac{1}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right) \left[1+\rho(\boldsymbol{x}_i,\boldsymbol{\mu}_j;\boldsymbol{\Sigma}_j)/\nu_j\right]^{\frac{\nu_j+p}{2}}} \right] \end{aligned} \tag{3.19}$$

In a Gaussian component mixture model-based approach, the complete data likelihood and log-likelihood functions are generated in a similar manner as the equations (3.18) and (3.19), respectively [4][5]. This is so because for a Gaussian component-based mixture model, the complete data set is

adequately specified by (3.13) where the observed sample data $\boldsymbol{x}_o$ is augmented only by the indicator latent data $\boldsymbol{z}$ [1]. However, with the current t component mixture model-based approach, the data vector $\boldsymbol{x}_{pc}$ comprising of the observed values $\boldsymbol{x}_o = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$ and the latent vectors $\boldsymbol{z} = \left(\boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T\right)^T$ is still considered to be an incomplete data set [3]. Further missing data need to be introduced into the mixture model.

In view of the characterization of the t-distribution in Theorem 2.2.3 and part (v) of Remark 2.2.1, further latent variables $\boldsymbol{W} = (W_1, \cdots, W_n)^T$ are introduced into the model (see [3] Section 3 and [19] Section 2), defined in such a way that given $z_{ij} = 1$;

$$\boldsymbol{X}_i | w_i, z_{ij} = 1 \sim N\left(\boldsymbol{\mu}_j, \frac{\boldsymbol{\Sigma}_j}{w_i}\right) \tag{3.20}$$

independently for $\quad i = 1, \cdots, n \quad$ and

$$W_i | z_{ij} = 1 \sim \text{ gamma } \left(\frac{\nu_j}{2}, \frac{\nu_j}{2}\right) \tag{3.21}$$

The complete data vector in finite mixture models that feature multivariate t-distributions is specified as $\boldsymbol{X}_c = \left(\boldsymbol{X}_o^T,\ \boldsymbol{Z}^T,\ \boldsymbol{W}^T\right)^T = \left(\boldsymbol{X}_{pc}^T,\ \boldsymbol{W}^T\right)^T$. Given our observed sample data $\boldsymbol{x}_o$, we have;

$$\boldsymbol{x_c} = \left(\boldsymbol{x}_{pc}^T,\ \boldsymbol{w}^T\right)^T = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T,\ \boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T,\ w_1, \cdots, w_n\right)^T \tag{3.22}$$

where $\boldsymbol{w}^T = (w_1, \cdots, w_n)$, denotes the latent weight variable. Thus, the complete latent vector is $\left(\boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_n^T, w_1, \cdots, w_n\right)^T$. We can generate the complete-data likelihood function associated with the complete data vector (3.22) and the corresponding complete log-likelihood functions by rewriting the likelihood function (3.18) as a product of the marginal densities of the $\boldsymbol{Z}_i$, the conditional densities of the W_i given $\boldsymbol{z}_i$ and the conditional density of $\boldsymbol{X}_i$ given w_i and $\boldsymbol{z}_i$ [3]. Thus, from (3.15), (2.18) and (2.19) we have

$$\begin{aligned}
L_c\left(\boldsymbol{\Psi};\ \boldsymbol{x}_c\right) = & \prod_{i=1}^{n}\prod_{j=1}^{g}\left[\tau_j \ \times \ \frac{\frac{\nu_j}{2}^{\frac{\nu_j}{2}}}{\Gamma\left(\frac{\nu_j}{2}\right)} w_i^{\frac{\nu_j}{2}-1} e^{-\frac{\nu_j}{2}w_i} \ \times \ \frac{1}{(2\pi)^{\frac{p}{2}}\left(\frac{1}{w_i}\right)^{\frac{p}{2}}|\boldsymbol{\Sigma}_j|^{\frac{1}{2}}}\right. \\
& \left. \times \ \exp\left\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T\left(\frac{\boldsymbol{\Sigma}_j}{w_i}\right)^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right\}\right]^{z_{ij}} \\
= & \prod_{i=1}^{n}\prod_{j=1}^{g}\left[\tau_j \frac{\frac{\nu_j}{2}^{\frac{\nu_j}{2}} w_i^{\frac{p}{2}}}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}_j|^{\frac{1}{2}}\ \Gamma\left(\frac{\nu_j}{2}\right)} w_i^{\frac{\nu_j}{2}-1} e^{-\frac{\nu_j}{2}w_i}\right. \\
& \left. \times \exp\left\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T\left(\frac{\boldsymbol{\Sigma}_j}{w_i}\right)^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right\}\right]^{z_{ij}} \\
= & \prod_{i=1}^{n}\prod_{j=1}^{g}\left[\tau_j \frac{\frac{\nu_j}{2}^{\frac{\nu_j}{2}} w_i^{\frac{p}{2}} w_i^{\frac{\nu_j}{2}-1} e^{-\frac{\nu_j}{2}w_i}}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}_j|^{\frac{1}{2}}\Gamma\left(\frac{\nu_j}{2}\right)} \exp\left\{-\frac{1}{2}\rho\left(\boldsymbol{x}_i, \boldsymbol{\mu}_j, \frac{\boldsymbol{\Sigma}_j}{w_i}\right)\right\}\right]^{z_{ij}}
\end{aligned}$$

where

$$\rho(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j/w_i) = (\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T (\boldsymbol{\Sigma}_j/w_i)^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j) \tag{3.23}$$

is the Mahalanobis distance from $\boldsymbol{x}_i$ to the center $\boldsymbol{\mu}_j$ with respect to the scale matrix $\frac{\boldsymbol{\Sigma}_j}{w_i}$. Therefore, the likelihood function of the complete data set $\boldsymbol{X}_c$ denoted by $L_c(\boldsymbol{\Psi}) = L_c(\boldsymbol{\Psi},\ \boldsymbol{x}_c)$ is given by

$$\begin{aligned} L_c(\boldsymbol{\Psi},\ \boldsymbol{x}_c) = \prod_{i=1}^{n}\prod_{j=1}^{g} \Bigg[\tau_j \frac{\frac{\nu_j}{2}^{\frac{\nu_j}{2}} w_i^{\frac{\nu_j+p}{2}-1} e^{-\frac{\nu_j w_i}{2}}}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_j|^{\frac{1}{2}} \Gamma\left(\frac{\nu_j}{2}\right)} \\ \times \exp\left\{ -\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \left(\frac{\boldsymbol{\Sigma}_j}{w_i}\right)^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j) \right\} \Bigg]^{z_{ij}} \end{aligned} \tag{3.24}$$

Taking the logarithm of the likelihood function (3.24) gives the complete-data log-likelihood function denoted by $l_c(\boldsymbol{\Psi}) = l_c(\boldsymbol{\Psi},\ \boldsymbol{x}_c)$, which can be written as:

$$l_c(\boldsymbol{\Psi}) = \sum_{j=1}^{g}\sum_{i=1}^{n} z_{ij} \left[\ln \tau_j + \ln \frac{w_i^{\frac{\nu_j+p}{2}-1} e^{-\frac{w_i\nu_j}{2} + \left\{-\frac{1}{2}\rho(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j/w_i)\right\}}}{\frac{2}{\nu_j}^{\frac{\nu_j}{2}} (2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_j|^{\frac{1}{2}} \Gamma\left(\frac{\nu_j}{2}\right)} \right] \tag{3.25}$$

It is important to recall that the observed data likelihood (2.21) is the function to be maximized, but the EM algorithm uses the "complete" data likelihood, (3.24), and its corresponding natural logarithm, (3.25) to find the ML estimates of the model parameter $\boldsymbol{\Psi} = \left(\boldsymbol{\tau}^T,\ \boldsymbol{\lambda}^T,\ \boldsymbol{\nu}^T\right)^T$.

Algebraic simplification of the complete data log-likelihood function (3.25) results $l_c(\boldsymbol{\Psi})$ being written as a sum of functions of $\boldsymbol{\tau}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$. To see this

$$\begin{aligned} l_c(\boldsymbol{\Psi}) &= \sum_{j=1}^{g}\sum_{i=1}^{n} z_{ij} \left[\ln \tau_j + \ln \frac{w_i^{\frac{\nu_j+p}{2}-1} e^{-\frac{w_i\nu_j}{2} + \left\{-\frac{1}{2}\rho(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j/w_i)\right\}}}{\frac{2}{\nu_j}^{\frac{\nu_j}{2}} (2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_j|^{\frac{1}{2}} \Gamma\left(\frac{\nu_j}{2}\right)} \right] \\ &= \sum_{j=1}^{g}\sum_{i=1}^{n} z_{ij} \Bigg[\ln \tau_j + \left(\frac{\nu_j + p}{2} - 1\right) \ln w_i - \frac{w_i\nu_j}{2} \\ &\qquad + \left\{ -\frac{1}{2}\rho(\boldsymbol{x}_i,\ \boldsymbol{\mu}_j;\ \boldsymbol{\Sigma}_j/w_i) \right\} - \frac{\nu_j}{2}\ln\frac{2}{\nu_j} \\ &\qquad - \frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| - \ln\Gamma\left(\frac{\nu_j}{2}\right) \Bigg] \\ &= \sum_{j=1}^{g}\sum_{i=1}^{n} z_{ij} \Bigg[\ln\tau_j - \ln\Gamma\left(\frac{\nu_j}{2}\right) + \frac{\nu_j}{2}\ln\frac{\nu_j}{2} + \frac{\nu_j}{2}(\ln w_i - w_i) \\ &\qquad + \frac{p}{2}\ln w_i - \ln w_i - \frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| - \frac{w_i}{2}\rho(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j) \Bigg] \end{aligned}$$

Thus, the complete data log-likelihood can be written as

$$l_c(\boldsymbol{\Psi}) \quad = \quad l_{c_1}(\boldsymbol{\tau}) + l_{c_2}(\boldsymbol{\lambda}) + l_{c_3}(\boldsymbol{\nu}) \tag{3.26}$$

where;

$$l_{c_1}(\boldsymbol{\tau}) = \sum_{j=1}^{g}\sum_{i=1}^{n} z_{ij} \ln \tau_j \quad , \tag{3.27}$$

$$\begin{aligned} l_{c_2}(\boldsymbol{\nu}) = \sum_{j=1}^{g}\sum_{i=1}^{n} z_{ij} \Big\{ &- \ln \Gamma\left(\frac{\nu_j}{2}\right) + \frac{\nu_j}{2} \ln \frac{\nu_j}{2} \\ &+ \frac{\nu_j}{2}(\ln w_i - w_i) - \ln w_i \Big\} \end{aligned} \tag{3.28}$$

and

$$\begin{aligned} l_{c_3}(\boldsymbol{\lambda}) = \sum_{j=1}^{g}\sum_{i=1}^{n} z_{ij} \Big\{ &- \frac{p}{2} \ln(2\pi) - \frac{1}{2} \ln |\boldsymbol{\Sigma}_j| \\ &+ \frac{p}{2} \ln w_i - \frac{w_i}{2} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) \Big\} \end{aligned} \tag{3.29}$$

The functions (3.27), (3.28) and (3.29) are used in the EM algorithm to maximize the complete data log-likelihood function (3.25). Maximizing (3.25) through maximizing the functions (3.27), (3.28) and (3.29) will maximize the observed data likelihood function (2.21) indirectly. The EM algorithm iterates between two steps which we discuss in detail in the next two sections of this chapter. Section (3.2.1) details the E-step of the EM algorithm as applied to finite mixtures of multivariate t-distributions. Section (3.2.2) gives the details of the M-step.

3.2.1 The Expectation-Step

Let $\boldsymbol{\Psi}^{(m)}$ denote the current value of $\boldsymbol{\Psi}$ after m iterations of the EM algorithm where

$$\boldsymbol{\Psi}^{(m)} = \left(\boldsymbol{\tau}^{(m)T},\ \boldsymbol{\lambda}^{(m)T},\ \boldsymbol{\nu}^{(m)T}\right)^T \qquad m \in \mathbb{N} \cup \{0\}$$

At the $(m+1)^{th}$ iteration, the E-step of the EM-algorithm computes the value of $Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)})$, the current conditional expectation of the complete data log-likelihood function, using the current parameter value $\boldsymbol{\Psi}^{(m)}$. Thus, the E-step can be written as:

$$\begin{aligned} Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)}) =& E_{\boldsymbol{\Psi}^{(m)}}\left[l_c(\boldsymbol{\Psi})\right] \\ =& E_{\boldsymbol{\Psi}^{(m)}}\left[l_{c1}(\boldsymbol{\tau}) + l_{c2}(\boldsymbol{\nu}) + l_{c3}(\boldsymbol{\lambda})\right] \end{aligned} \tag{3.30}$$

In practice, computing the value of (3.30) is achieved by computing the following quantities, for all $i = 1, \cdots, n$ and for all $j = 1, \cdots, g$:

i) The value of the conditional expectation of each of the Z_{ij} given $\boldsymbol{x}_o$:

$$e_{ij}^{(m)} = E_{\boldsymbol{\Psi}^{(m)}}\left[Z_{ij}|\boldsymbol{x}_i\right]$$

ii) The value of the conditional expectation of W_i given $\boldsymbol{x}_i$ and z_{ij}:

$$\sigma_{ij}^{(m)} = E_{\boldsymbol{\Psi}^{(m)}}\left[W_i|\boldsymbol{x}_i, z_{ij}=1\right]$$

iii) The value of the conditional expectation of $\ln W_i$ given $\boldsymbol{x}_i$ and z_{ij}:

$$\xi_{ij}^{(m)} = E_{\boldsymbol{\Psi}^{(m)}}\left[\ln W_i|\boldsymbol{x}_i, z_{ij}=1\right]$$

The conditional expected value of Z_{ij} can be found using the definition of expected value, Bayes' Theorem and the probability density functions (3.14) and (3.16).

$$\begin{aligned}
E_{\boldsymbol{\Psi}^{(m)}}\left[Z_{ij}|\boldsymbol{x}_i\right] =& \sum_{z_{ij}} z_{ij}.Pr(Z_{ij}=z_{ij}|\ \boldsymbol{x}_i) \\
=& 1.Pr(Z_{ij}=1|\ \boldsymbol{x}_i) + 0.Pr(Z_{ij}=0|\ \boldsymbol{x}_i) \\
=& Pr(Z_{ij}=1|\ \boldsymbol{x}_i) \\
=& \frac{f(\boldsymbol{x}_i|\ Z_{ij}=1).Pr(Z_{ij}=1)}{f(\boldsymbol{x}_i)} \\
=& \frac{f(\boldsymbol{x}_i|\ Z_{ij}=1, \boldsymbol{\Psi}^{(m)}).Pr(Z_{ij}=1|\ \boldsymbol{\Psi}^{(m)})}{f(\boldsymbol{x}_i|\ \boldsymbol{\Psi}^{(m)})} \\
=& \frac{f\left(\boldsymbol{x}_i|\boldsymbol{\theta}_j^{(m)}\right).\tau_j^{(m)}}{\sum_{k=1}^{g}\tau_k^{(m)} f\left(\boldsymbol{x}_i|\boldsymbol{\theta}_k^{(m)}\right)}
\end{aligned}$$

Therefore, at the $(m+1)^{th}$ iteration, the E-Step evaluates $e_{ij}^{(m)}$ as:

$$\begin{aligned}
e_{ij}^{(m)} =& E_{\boldsymbol{\Psi}^{(m)}}\left[Z_{ij}|\boldsymbol{x}_i\right] \\
=& \frac{\tau_j^{(m)} f\left(\boldsymbol{x}_i|\boldsymbol{\theta}_j^{(m)}\right)}{\sum_{k=1}^{g}\tau_k^{(m)} f\left(\boldsymbol{x}_i|\boldsymbol{\theta}_k^{(m)}\right)}
\end{aligned} \tag{3.31}$$

which is just the posterior probability that $\boldsymbol{x}_i$ belongs to the j^{th} component of the mixture model, using the current value $\boldsymbol{\Psi}^{(m)}$ of $\boldsymbol{\Psi}$.

To compute the value of $\sigma_{ij}^{(m)}$, we use the fact that the prior probability distribution of W_i given $z_{ij}=1$ is a gamma distribution with scale parameter $\lambda = \frac{\nu_j^{(m)}}{2}$ and slope parameter $\alpha = \frac{\nu_j^{(m)}}{2}$ [see (3.21) and (2.3)]. Given the observed data set $\boldsymbol{x}_o$, we determine the posterior conditional probability distribution of W_i using Baye's theorem as follows:

$$\begin{aligned}
f(w_i|\ \boldsymbol{x}_i,\ z_{ij}=1) =& \frac{f(w_i, \boldsymbol{x}_i, z_{ij}=1)}{f(\boldsymbol{x}_i, z_{ij}=1)} \\
=& \frac{f(\boldsymbol{x}_i|\ w_i, z_{ij}=1) f(w_i, z_{ij}=1)}{f(\boldsymbol{x}_i|\ z_{ij}=1) f(z_{ij}=1)} \\
=& \frac{f(\boldsymbol{x}_i|\ w_i, z_{ij}=1) f(w_i|\ z_{ij}=1)}{f(\boldsymbol{x}_i|\ z_{ij}=1)}
\end{aligned} \tag{3.32}$$

From equations (2.18), and (2.19) of Remark 2.2.1, we have

$$f(w_i|\ z_{ij}=1)=\frac{\frac{\nu_j^{(m)}}{2}^{\frac{\nu_j^{(m)}}{2}}}{\Gamma\left(\frac{\nu_j^{(m)}}{2}\right)}w_i^{\frac{\nu_j^{(m)}}{2}-1}e^{-\frac{\nu_j^{(m)}}{2}w_i} \tag{3.33}$$

$$f\left(\boldsymbol{x}_i|\ w_i,z_{ij}=1\right)=\frac{w^{\frac{p}{2}}}{(2\pi)^{\frac{p}{2}}\left|\boldsymbol{\Sigma}_j^{(m)}\right|^{\frac{1}{2}}}.\exp\left\{-\frac{w_i}{2}\rho(\boldsymbol{x}_i,\boldsymbol{\mu}_j^{(m)};\boldsymbol{\Sigma}_j^{(m)})\right\} \tag{3.34}$$

and from Definition 2.1.1, we have

$$f(\boldsymbol{x}_i|\ z_{ij}=1)=\frac{\Gamma\left(\frac{\nu_j^{(m)}+p}{2}\right)\left|\boldsymbol{\Sigma}_j^{(m)}\right|^{-\frac{1}{2}}}{(\pi\nu_j^{(m)})^{\frac{p}{2}}\Gamma\left(\frac{\nu_j^{(m)}}{2}\right)\left[1+\rho(\boldsymbol{x}_i,\boldsymbol{\mu}_j^{(m)};\boldsymbol{\Sigma}_j^{(m)})/\nu_j^{(m)}\right]^{\frac{\nu_j^{(m)}+p}{2}}} \tag{3.35}$$

Substituting (3.33), (3.34) and (3.35) into (3.32) and simplifying, we have

$$\begin{aligned}f(w_i|\ \boldsymbol{x}_i,\ z_{ij}=1)=&\frac{\left[\frac{1}{2}\left(\nu_j^{(m)}+\rho(\boldsymbol{x}_i,\boldsymbol{\mu}_j^{(m)};\boldsymbol{\Sigma}_j^{(m)})\right)\right]^{\frac{\nu_j^{(m)}+p}{2}}}{\Gamma\left(\frac{\nu_j^{(m)}+p}{2}\right)}w^{\frac{\nu_j^{(m)}+p}{2}-1}\\&\times\exp\left\{-\frac{1}{2}\left(\nu_j^{(m)}+\rho(\boldsymbol{x}_i,\boldsymbol{\mu}_j^{(m)};\boldsymbol{\Sigma}_j^{(m)})\right)w_i\right\}\end{aligned} \tag{3.36}$$

which is the probability density function of a gamma distribution with parameters $\alpha_f^{(m)}$ and $\lambda_f^{(m)}$ given by:

$$\alpha_f^{(m)}=\frac{\nu_j^{(m)}+p}{2}\quad\text{and}\quad\lambda_f^{(m)}=\frac{\nu_j^{(m)}+\rho\left(\boldsymbol{x}_i,\boldsymbol{\mu}_j^{(m)};\ \boldsymbol{\Sigma}_j^{(m)}\right)}{2} \tag{3.37}$$

Therefore, at the $(m+1)^{th}$ iteration, the posterior conditional probability distribution of W_i is a gamma distribution with parameters $\alpha_f^{(m)}$ and $\lambda_f^{(m)}$ [3] so that the expectation of W_i conditioned on $\boldsymbol{x}_i$ and z_{ij} when the current parameter value is $\boldsymbol{\Psi}^{(m)}$, is given by:

$$\begin{aligned}\sigma_{ij}^{(m)}=&E_{\boldsymbol{\Psi}^{(m)}}\left[W_i|\boldsymbol{x}_i,z_{ij}\right]\\=&\frac{\alpha_f^{(m)}}{\lambda_f^{(m)}}\\=&\frac{\nu_j^{(m)}+p}{\nu_j^{(m)}+\rho\left(\boldsymbol{x}_i,\boldsymbol{\mu}_j^{(m)};\ \boldsymbol{\Sigma}_j^{(m)}\right)}\\=&\frac{\nu_j^{(m)}+p}{\nu_j^{(m)}+\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{(m)}\right)^T\boldsymbol{\Sigma}^{-1(m)}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{(m)}\right)}\end{aligned} \tag{3.38}$$

To determine $\xi_{ij}^{(m)}$, the expectation of $\ln W_i$ conditioned on $\boldsymbol{x}_i$ and z_{ij}, we recall the digamma function:

Definition 3.2.1. *[20] The digamma function, denoted by $\psi(w)$, is defined as the logarithmic derivative of the gamma function $\Gamma(w)$*

$$\psi(w) = \left[\frac{\partial \Gamma(w)}{\partial(w)}\right]\left[\frac{1}{\Gamma(w)}\right] \tag{3.39}$$

Theorem 3.2.2. *If $W \sim Gamma(\alpha,\ \lambda)$, where α is the shape parameter and λ the rate parameter, then*

$$\mathbb{E}(\ln W) = \psi(\alpha) - \ln \lambda$$

Proof. The results here uses the fact that if a distribution is a member of the natural exponential family of distributions, the expected value of the sufficient statistic can be obtained from the first order derivative of the log-partition function. The gamma distribution has a natural exponential distribution:

$$\begin{aligned} f(w) =& \frac{(\lambda)^{\alpha}\ w^{\alpha-1}}{\Gamma(\alpha)} e^{-\lambda\ w} \\ =& \frac{\lambda^{\alpha}}{\Gamma(\alpha)} . \exp\left\{(\alpha-1)\ln w - \lambda\ w\right\} .1 \end{aligned} \tag{3.40}$$

Let $\boldsymbol{\eta} = (\eta_1,\ \eta_2)^T$ denote the natural parameter where $\eta_1 = \alpha - 1$ and $\eta_2 = -\lambda$. Then $\alpha = \eta_1 + 1$ and $\lambda = -\eta_2$. We obtain the canonical form of the density function as

$$\begin{aligned} f(w) =& \frac{(-\eta_2)^{(\eta_1+1)}}{\Gamma(\eta_1+1)} . \exp\left\{\eta_1 \ln w + \eta_2\ w\right\} .1 \\ =& \exp\left\{\eta_1 \ln w + \eta_2\ w\ -\ \ln\left[\frac{(-\eta_2)^{(\eta_1+1)}}{\Gamma(\eta_1+1)}\right]\right\} .1 \\ =& \exp\left\{\eta_1 \ln w + \eta_2\ w\right. \\ & \left. -\ \left[\ln\Gamma(\eta_1+1) - (\eta_1+1)\ln(-\eta_2)\right]\right\} .1 \end{aligned} \tag{3.41}$$

From (3.41), we see that the sufficient statistics for the gamma distribution are $T_1(w) = \ln w$ and $T_2(w) = w$, and the log-partition function is

$$C(\boldsymbol{\eta}) = \ln\Gamma(\eta_1+1) - (\eta_1+1)\ln(-\eta_2) \tag{3.42}$$

so that

$$\mathbb{E}(\ln W) = \frac{\partial\ C(\boldsymbol{\eta})}{\partial \eta_1} \quad \text{and} \quad \mathbb{E}(W) = \frac{\partial\ C(\boldsymbol{\eta})}{\partial \eta_2}$$

Thus,

$$\begin{aligned} \mathbb{E}(\ln W) =& \frac{1}{\Gamma(\eta_1+1)} . \frac{\partial}{\partial\ w}\Gamma(\eta_1+1) - \ln(-\eta_2) \\ =& \frac{1}{\Gamma(\alpha)} . \frac{\partial}{\partial\ w}\Gamma(\alpha) - \ln(\lambda) \\ =& \psi(\alpha) - \ln(\lambda) \end{aligned}$$

□

In the E-step, the posterior conditional distribution of W_i has parameters $\alpha_f^{(m)}$ and $\lambda_f^{(m)}$ as defined in (3.37). Using Theorem 3.2.2, the conditional expectation of $\ln W_i$ given $\boldsymbol{x}_i$, z_{ij} and the current parameter value $\boldsymbol{\Psi^{(m)}}$ is

$$\begin{aligned}
\xi_{ij}^{(m)} =& E_{\boldsymbol{\Psi^{(m)}}}\left[\ln W_i | \boldsymbol{x}_i, z_{ij}=1\right] \\
=& \psi(\alpha_f^{(m)}) - \ln(\lambda_f^{(m)}) \\
=& \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left[\frac{\nu_j^{(m)}+\rho\left(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(m)};\ \boldsymbol{\Sigma}_j^{(m)}\right)}{2}\right] \\
=& \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) + \ln\left[\frac{2}{\nu_j^{(m)}+\rho\left(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(m)};\ \boldsymbol{\Sigma}_j^{(m)}\right)} . \frac{\nu_j^{(m)}+p}{\nu_j^{(m)}+p}\right] \\
=& \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) + \ln\left[\frac{\nu_j^{(m)}+p}{\nu_j^{(m)}+\rho\left(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(m)};\ \boldsymbol{\Sigma}_j^{(m)}\right)}\right] - \ln\left[\frac{\nu_j^{(m)}+p}{2}\right] \\
=& \ln\left[\frac{\nu_j^{(m)}+p}{\nu_j^{(m)}+\rho\left(\boldsymbol{x}_i, \boldsymbol{\mu}_j^{(m)};\ \boldsymbol{\Sigma}_j^{(m)}\right)}\right] + \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left[\frac{\nu_j^{(m)}+p}{2}\right] \\
=& \ln\left(\sigma_{ij}^{(m)}\right) + \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left(\frac{\nu_j^{(m)}+p}{2}\right) \qquad (3.43)
\end{aligned}$$

To summarize the E-step, we see that this step computes the updated values of the latent variables ($\boldsymbol{Z}$, W, $\ln W$) as presented in (3.31), (3.38) and (3.43). Therefore, the conditional expectation of the complete data log-likelihood function $Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)})$, can now be written as:

$$Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(m)}) = Q_1(\boldsymbol{\tau}|\boldsymbol{\Psi}^{(m)}) + Q_2(\boldsymbol{\nu}|\boldsymbol{\Psi}^{(m)}) + Q_3(\boldsymbol{\lambda}|\boldsymbol{\Psi}^{(m)}) \qquad (3.44)$$

where;

$$Q_1(\boldsymbol{\tau}|\boldsymbol{\Psi}^{(m)}) = \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \ln \tau_j \qquad (3.45)$$

$$\begin{aligned}
Q_2(\boldsymbol{\nu}|\boldsymbol{\Psi}^{(m)}) =& \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \left[-\ln\Gamma\left(\frac{\nu_j}{2}\right) + \frac{\nu_j}{2}\ln\frac{\nu_j}{2}\right. \\
& \left. - \ln\sigma_{ij}^{(m)} + \frac{\nu_j}{2}\left(\xi_{ij}^{(m)} - \sigma_{ij}^{(m)}\right)\right] \\
=& \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \left[-\ln\Gamma\left(\frac{\nu_j}{2}\right) + \frac{\nu_j}{2}\ln\frac{\nu_j}{2}\right. \\
& - \ln\sigma_{ij}^{(m)} + \frac{\nu_j}{2}\left\{\left(\ln\sigma_{ij}^{(m)} - \sigma_{ij}^{(m)}\right)\right. \\
& \left.\left. + \ \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left(\frac{\nu_j^{(m)}+p}{2}\right)\right\}\right] \qquad (3.46)
\end{aligned}$$

and

$$Q_3(\boldsymbol{\lambda}|\boldsymbol{\Psi}^{(m)}) = \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \left[-\frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| + \frac{p}{2}\ln\sigma_{ij}^{(m)} - \frac{\sigma_{ij}^{(m)}}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) \right] \quad (3.47)$$

According to Section 7 of [3], the term $\left(\ln\sigma_{ij}^{(m)} - \sigma_{ij}^{(m)}\right)$ in the expression (3.46) can be substituted with the weighted term given by

$$\frac{1}{n_i^{(m)}}\sum_{i=1}^{n} e_{ij}^{(m)}\left(\ln\sigma_{ij}^{(m)} - \sigma_{ij}^{(m)}\right)$$

where

$$n_i^{(m)} = \sum_{i=1}^{n} e_{ij}^{(m)}$$

so that the function $Q_2(\boldsymbol{\nu}|\boldsymbol{\Psi}^{(m)})$ from (3.46) can be written as

$$Q_2(\boldsymbol{\nu}|\boldsymbol{\Psi}^{(m)}) = \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \left[-\ln\Gamma\left(\frac{\nu_j}{2}\right) + \frac{\nu_j}{2}\ln\frac{\nu_j}{2} - \ln\sigma_{ij}^{(m)} + \frac{\nu_j}{2}\left\{ \frac{1}{n_i^{(m)}}\sum_{i=1}^{n} e_{ij}^{(m)}\left(\ln\sigma_{ij}^{(m)} - \sigma_{ij}^{(m)}\right) + \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left(\frac{\nu_j^{(m)}+p}{2}\right)\right\}\right] \quad (3.48)$$

3.2.2 The Maximization-Step

The E-step at the $(m+1)^{th}$ iteration gives an updated estimate of the latent variables $(\boldsymbol{Z},\ W,\ \ln W)$. These estimates are then used to update the model parameter $\boldsymbol{\Psi}$. Let $Q^{(m)}$, be defined by

$$Q^{(m)}\left(\boldsymbol{\Psi}\right) = Q_1^{(m)}(\boldsymbol{\tau}) + Q_2^{(m)}(\boldsymbol{\nu}) + Q_3^{(m)}(\boldsymbol{\lambda}) \quad (3.49)$$

where;

$$Q_1^{(m)}(\boldsymbol{\tau}) = \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)}\ln\tau_j \quad (3.50)$$

$$Q_2^{(m)}(\boldsymbol{\nu}) = \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \left[-\ln\Gamma\left(\frac{\nu_j}{2}\right) + \frac{\nu_j}{2}\ln\frac{\nu_j}{2} - \ln\sigma_{ij}^{(m)} + \frac{\nu_j}{2}\left\{ \frac{1}{n_i^{(m)}}\sum_{i=1}^{n} e_{ij}^{(m)}\left(\ln\sigma_{ij}^{(m)} - \sigma_{ij}^{(m)}\right) + \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left(\frac{\nu_j^{(m)}+p}{2}\right)\right\}\right] \quad (3.51)$$

and

$$Q_3^{(m)}(\boldsymbol{\lambda}) = \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \left[-\frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| + \frac{p}{2}\ln\sigma_{ij}^{(m)} - \frac{\sigma_{ij}^{(m)}}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) \right] \tag{3.52}$$

On the M-step, the quantity $Q^{(m)}(\boldsymbol{\Psi})$ is maximized with respect to the group membership probability parameter, $\boldsymbol{\tau}$ and the component parameters, $\boldsymbol{\theta} = \left(\boldsymbol{\lambda}^T, \boldsymbol{\nu}^T\right)^T$ to produce

$$\boldsymbol{\Psi}^{(m+1)} = \left(\boldsymbol{\tau}^{(m+1)T},\ \boldsymbol{\lambda}^{(m+1)T},\ \boldsymbol{\nu}^{(m+1)T}\right)^T$$

The parameters $\boldsymbol{\tau}$, $\boldsymbol{\nu}$ and $\boldsymbol{\lambda}$ are updated independently using (3.50), (3.51) and (3.52), respectively [3]. The parameter for the component mixing proportion $\boldsymbol{\tau}$ is updated by solving the equation:

$$\begin{aligned} \boldsymbol{\tau}^{(m+1)} &= \arg\max_{\boldsymbol{\tau}}\ Q^{(m)}\left(\boldsymbol{\Psi}\right) \\ &= \arg\max_{\boldsymbol{\tau}}\ Q_1^{(m)}\left(\boldsymbol{\tau}\right) \\ &= \arg\max_{\boldsymbol{\tau}}\ \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \ln\tau_j \end{aligned} \tag{3.53}$$

This is achieved by updating each τ_j for $j = 1, \cdots, g$ as the average of the posterior probabilities of component memberships of the mixture model [1]:

$$\tau_j^{(m+1)} = \frac{1}{n}\sum_{i=1}^{n} e_{ij}^{(m)} \qquad \text{for } j = 1, \cdots, g \tag{3.54}$$

In order to update the location parameters $\boldsymbol{\mu}_j$ and the scale parameters $\boldsymbol{\Sigma}_j$ for $j = 1, \cdots, g$, we solve the equation

$$\begin{aligned} \boldsymbol{\lambda}^{(m+1)} &= \arg\max_{\boldsymbol{\lambda}}\ Q^{(m)}\left(\boldsymbol{\Psi}\right) \\ &= \arg\max_{\boldsymbol{\lambda}}\ Q_3^{(m)}\left(\boldsymbol{\lambda}\right) \\ &= \arg\max_{\boldsymbol{\mu},\ \boldsymbol{\Sigma}}\ \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \left[-\frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| \right. \\ &\qquad \left. + \frac{p}{2}\ln\sigma_{ij}^{(m)} - \frac{\sigma_{ij}^{(m)}}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) \right] \end{aligned} \tag{3.55}$$

This is achieved by updating each center parameter $\boldsymbol{\mu}_j$ and each correlation matrix $\boldsymbol{\Sigma}_j$ for $j = 1, \cdots, g$. Maximizing the center $\boldsymbol{\mu}_j$ for the j^{th} component, we have

$$Q_{3j}^{(m)}\left(\boldsymbol{\lambda}\right) = \sum_{i=1}^{n} e_{ij}^{(m)} \left[-\frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| + \frac{p}{2}\ln\sigma_{ij}^{(m)} - \frac{\sigma_{ij}^{(m)}}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) \right]$$

so that ignoring the constants (terms not involving $\boldsymbol{\mu}_j$), we have the derivative

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\mu}_j} Q_{3j}^{(m)}(\boldsymbol{\lambda}) &= \sum_{i=1}^{n} e_{ij}^{(m)} \left[-\frac{\partial}{\partial \boldsymbol{\mu}_j} \left\{ \frac{\sigma_{ij}^{(m)}}{2} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) \right\} \right] \\
&= \sum_{i=1}^{n} \frac{e_{ij}^{(m)} \sigma_{ij}^{(m)}}{2} \left[-\frac{\partial}{\partial \boldsymbol{\mu}_j} \left\{ \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right) \right\} \right] \\
&= \sum_{i=1}^{n} \frac{e_{ij}^{(m)} \sigma_{ij}^{(m)}}{2} \left[-\frac{\partial}{\partial \boldsymbol{\mu}_j} \left\{ \boldsymbol{x}_i^T \boldsymbol{\Sigma}_j^{-1} \boldsymbol{x}_i - 2\boldsymbol{x}_i^T \boldsymbol{\Sigma}_j^{-1} \boldsymbol{\mu}_j + \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}_j^{-1} \boldsymbol{\mu}_j \right\} \right] \\
&= \sum_{i=1}^{n} \frac{e_{ij}^{(m)} \sigma_{ij}^{(m)}}{2} \left[-2\boldsymbol{x}_i^T \boldsymbol{\Sigma}_j^{-1} + 2\boldsymbol{\mu}_j^T \boldsymbol{\Sigma}_j^{-1} \right] \\
&= \sum_{i=1}^{n} e_{ij}^{(m)} \sigma_{ij}^{(m)} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}_j^{-1} \; - \; \sum_{i=1}^{n} e_{ij}^{(m)} \sigma_{ij}^{(m)} \boldsymbol{x}_i^T \boldsymbol{\Sigma}_j^{-1}
\end{aligned}$$

Setting the derivative $\partial\, Q_{3j}^{(m)}(\boldsymbol{\lambda}) \Big/ \partial \boldsymbol{\mu}_j = 0$, we have

$$\hat{\boldsymbol{\mu}}_j^T = \frac{\sum_{i=1}^{n} e_{ij}^{(m)} \sigma_{ij}^{(m)} \boldsymbol{x}_i^T}{\sum_{i=1}^{n} e_{ij}^{(m)} \sigma_{ij}^{(m)}} \tag{3.56}$$

Maximizing $Q_{3j}^{(m)}(\boldsymbol{\lambda})$ with respect to $\boldsymbol{\Sigma}_j$ gives the ML estimate of $\boldsymbol{\Sigma}_j$, the scale matrix for the j^{th} component. Ignoring constants (terms not involving $\boldsymbol{\Sigma}_j$), differentiating $Q_{3j}^{(m)}(\boldsymbol{\lambda})$ gives

$$\frac{\partial\, Q_{3j}^{(m)}(\boldsymbol{\lambda})}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)} = \sum_{i=1}^{n} e_{ij}^{(m)} \left[-\frac{\partial}{\partial \boldsymbol{\Sigma}_j^{-1}} \left\{ \frac{1}{2} \ln |\boldsymbol{\Sigma}_j| + \frac{\sigma_{ij}^{(m)}}{2} \rho(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j) \right\} \right]$$

$$
\begin{aligned}
\frac{\partial\, Q_{3j}^{(m)}(\boldsymbol{\lambda})}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)} &= \sum_{i=1}^{n} \frac{e_{ij}^{(m)}}{2}\left[\frac{\partial \ln\left|\boldsymbol{\Sigma}_j^{-1}\right|}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)}\right] - \sum_{i=1}^{n} \frac{e_{ij}^{(m)}\sigma_{ij}^{(m)}}{2}\left[\frac{\partial\left\{\rho(\boldsymbol{x}_i, \boldsymbol{\mu}_j; \boldsymbol{\Sigma}_j)\right\}}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)}\right] \\
&= \sum_{i=1}^{n} \frac{e_{ij}^{(m)}}{2}\left[\frac{\partial}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)} \ln\left|\boldsymbol{\Sigma}_j^{-1}\right|\right] - \sum_{i=1}^{n} \frac{e_{ij}^{(m)}\sigma_{ij}^{(m)}}{2}\Bigg[\\
&\qquad \times \frac{\partial}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)}\left\{\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\right\}\Bigg] \\
&= \sum_{i=1}^{n} \frac{e_{ij}^{(m)}}{2}\left[\frac{\partial}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)} \ln\left|\boldsymbol{\Sigma}_j^{-1}\right|\right] - \sum_{i=1}^{n} \frac{e_{ij}^{(m)}\sigma_{ij}^{(m)}}{2}\Bigg[\\
&\qquad \times \frac{\partial}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)} \operatorname{trace}\left\{\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T\right\}\Bigg] \\
&= \sum_{i=1}^{n} \frac{e_{ij}^{(m)}}{2}\left[\frac{\partial}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)} \ln\left|\boldsymbol{\Sigma}_j^{-1}\right|\right] - \sum_{i=1}^{n} \frac{e_{ij}^{(m)}\sigma_{ij}^{(m)}}{2}\Bigg[\\
&\qquad \times \frac{\partial}{\partial\left(\boldsymbol{\Sigma}_j^{-1}\right)} \operatorname{trace}\left\{\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T\right\}\Bigg] \\
&= \sum_{i=1}^{n} \frac{e_{ij}^{(m)}}{2}\boldsymbol{\Sigma}_j - \sum_{i=1}^{n} \frac{e_{ij}^{(m)}\sigma_{ij}^{(m)}}{2}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j\right)^T
\end{aligned}
$$

Setting the derivative $\partial\, Q_{3j}^{(m)}(\boldsymbol{\lambda}) \Big/ \partial\left(\boldsymbol{\Sigma}_j^{-1}\right) = 0$ and solving for $\boldsymbol{\Sigma}_j$, we have

$$
\hat{\boldsymbol{\Sigma}}_j = \frac{\sum_{i=1}^{n} e_{ij}^{(m)}\sigma_{ij}^{(m)}\left(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j\right)\left(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j\right)^T}{\sum_{i=1}^{n} e_{ij}^{(m)}} \tag{3.57}
$$

Therefore, the solution to equation (3.55) which gives the updated center parameter $\boldsymbol{\mu}_j$ and scale parameter $\boldsymbol{\Sigma}_j$ for the j^{th} component is :

$$
\boldsymbol{\mu}_j^{(m+1)} = \frac{1}{\boldsymbol{n_i}} \sum_{i=1}^{n} e_{ij}^{(m)}\sigma_{ij}^{(m)}\boldsymbol{x_i} \tag{3.58}
$$

and

$$
\boldsymbol{\Sigma}_j^{(m+1)} = \frac{1}{n_i} \sum_{i=1}^{n} e_{ij}^{(m)}\sigma_{ij}^{(m)}(\boldsymbol{x_i} - \boldsymbol{\mu}_j^{(m+1)})(\boldsymbol{x_i} - \boldsymbol{\mu}_j^{(m+1)})^T \tag{3.59}
$$

where

$$
\boldsymbol{n_j} = \sum_{i=1}^{n} e_{ij}^{(m)}\sigma_{ij}^{(m)} \qquad \text{and} \qquad n_j = \sum_{i=1}^{n} e_{ij}^{(m)} \qquad \text{for } j = 1, \cdots, g
$$

This is equivalent to computing the weighted sample mean and sample covariance matrix of the sample data, $\boldsymbol{x}_o = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ with weights $w = \{w_1, \cdots, w_n\}$ for each $j = 1, \cdots, g$. Thus, this process is seen as the weighted least square estimation. In the E-step, the values of the weights W_i

are updated while in the M-step, the new values for the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are estimated using the weighted least square estimation process [1][3].

On the $(m+1)^{th}$ iteration, the M-step updates the value of the degrees of freedom parameter, $\boldsymbol{\nu}$ by solving the equation

$$\begin{aligned}
\boldsymbol{\nu}^{(m+1)} &= \arg\max_{\boldsymbol{\nu}} \; Q^{(m)}\left(\boldsymbol{\Psi}\right) \\
&= \arg\max_{\boldsymbol{\nu}} \; Q_2^{(m)}\left(\boldsymbol{\nu}\right) \\
&= \arg\max_{\boldsymbol{\nu}} \; \sum_{i=1}^{n}\sum_{j=1}^{g} e_{ij}^{(m)} \Big[-\ln\Gamma\left(\frac{\nu_j}{2}\right) + \frac{\nu_j}{2}\ln\frac{\nu_j}{2} \\
&\qquad - \ln\sigma_{ij}^{(m)} + \frac{\nu_j}{2}\left\{ \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} e_{ij}^{(m)} \left(\ln\sigma_{ij}^{(m)} - \sigma_{ij}^{(m)} \right) \right. \\
&\qquad \left. + \; \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left(\frac{\nu_j^{(m)}+p}{2}\right) \right\} \Big]
\end{aligned} \tag{3.60}$$

The degrees of freedom parameter $\boldsymbol{\nu}$, can be fixed at a given value for each of the g components. In this case, the component parameters to be estimated are restricted to the location parameter $\boldsymbol{\mu}$ and the correlation matrix $\boldsymbol{\Sigma}$. This approach simplifies the estimation process as the ML estimation of parameters exist in closed form [19]. The advantage of using the t-distribution over the use of Gaussian distributions is that the degrees of robustness as controlled by the parameter $\boldsymbol{\nu}$, can be inferred from the data by computing its maximum likelihood estimate [8].

The M-step for $\boldsymbol{\nu}$ however, is difficult because the solution to (3.60) must be obtained by finding the solution to the equation

$$\sum_{i=1}^{n} \frac{\partial}{\partial\nu_j} \, Q_{2j}\left(\nu_j \,\middle|\, \boldsymbol{\Psi}^{(m)}\right) = 0 \tag{3.61}$$

where on ignoring terms not involving ν from (3.60), we have

$$\begin{aligned}
Q_{2j}\left(\nu_j \,\middle|\, \boldsymbol{\Psi}^{(m)}\right) = &-\ln\Gamma\left(\frac{\nu_j}{2}\right) + \frac{\nu_j}{2}\ln\frac{\nu_j}{2} \\
&+ \frac{\nu_j}{2}\left\{ \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} e_{ij}^{(m)} \left(\ln\sigma_{ij}^{(m)} - \sigma_{ij}^{(m)} \right) \right. \\
&\left. + \; \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left(\frac{\nu_j^{(m)}+p}{2}\right) \right\}
\end{aligned} \tag{3.62}$$

The derivative in (3.61) can be written as

$$\begin{aligned}
\left\{ -\psi\left(\frac{\nu_j}{2}\right) + \ln\left(\frac{\nu_j}{2}\right) + 1 + \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} e_{ij}^{(m)} \left(\ln\sigma_{ij}^{(m)} - \sigma_{ij}^{(m)} \right) \right. \\
\left. + \; \psi\left(\frac{\nu_j^{(m)}+p}{2}\right) - \ln\left(\frac{\nu_j^{(m)}+p}{2}\right) \right\} \;\; = \;\; 0
\end{aligned} \tag{3.63}$$

where

$$n_j^{(m)} = \sum_{i=1}^{n} e_{ij}^{(m)}$$

The solution to the equation (3.63) gives the updated value of the parameter ν_j, which is $\nu_j^{(m+1)}$.

To determine the value of $\nu_j^{(m+1)}$, we let the derivative in (3.63) be denoted by the function $d_{(m)}(\nu)$. Then we have

$$d_{(m)}(\nu) = -\psi\left(\frac{\nu_j}{2}\right) + \ln\left(\frac{\nu_j}{2}\right) + D^{(m)}$$

where

$$D^{(m)} = 1 + \frac{1}{n_i^{(m)}} \sum_{i=1}^{n} e_{ij}^{(m)} \left(\ln \sigma_{ij}^{(m)} - \sigma_{ij}^{(m)}\right) + \psi\left(\frac{\nu_j^{(m)} + p}{2}\right) - \ln\left(\frac{\nu_j^{(m)} + p}{2}\right) \tag{3.64}$$

Proposition 3.2.3. *([19, Proposition 1]) Let $d_{(m)}(\nu)$ denote the derivative function defined as*

$$d_{(m)}(\nu) = -\psi\left(\frac{\nu_j}{2}\right) + \ln\left(\frac{\nu_j}{2}\right) + D^{(m)}$$

where $D^{(m)}$ is as defined in equation (3.64). Then the following hold

(i) $d_{(m)}(\nu)$ is concave over $(0,\ \infty)$ so that ${d_{(m)}}''(\nu) < 0 \quad \forall \nu \in (0,\ \infty)$

(ii) $d_{(m)}(\nu) - D^{(m)}$ is strictly decreasing over the interval $(0,\ \infty)$ with

$$\lim_{\nu \to 0^+} \left[d_{(m)}(\nu) - D^{(m)}\right] = \infty \quad \text{and} \quad \lim_{\nu \to \infty} \left[d_{(m)}(\nu) - D^{(m)}\right] = 0$$

(iii) $D^{(m)} \leq 0$ for all $\nu \in (0,\ \infty)$ with $D^{(m)} = 0$ if and only if $\nu = \infty$

Proof. See the proof to proposition 1 in [19]. □

For more details on the digamma function as used in equation (3.63) and its solution, the reader is referred to the discussion in the Appendix of [19]. Here, it suffices to note from Proposition 3.2.3 that the function $d_{(m)}(\nu)$ has a unique critical point in the interval $(0,\ \infty)$ so that the equation

$$\begin{aligned} \left\{ -\psi\left(\frac{\nu_j}{2}\right) + \ln\left(\frac{\nu_j}{2}\right) + 1 + \frac{1}{n_i^{(m)}} \sum_{i=1}^{n} e_{ij}^{(m)} \left(\ln \sigma_{ij}^{(m)} - \sigma_{ij}^{(m)}\right) \right. \\ \left. + \psi\left(\frac{\nu_j^{(m)} + p}{2}\right) - \ln\left(\frac{\nu_j^{(m)} + p}{2}\right) \right\} \quad = \quad 0 \end{aligned}$$

has a unique solution in the interval $(0,\ \infty)$, which is determined by ${d_{(m)}}'(\nu) = 0$ when $D^{(m)} < 0$. When $D^{(m)} = 0$, the maximization of $d_{(m)}(\nu)$ is given by

$$\sup_{\nu \in (0,\ \infty)} d_{(m)}(\nu) = \lim_{\nu \to \infty} d_{(m)}(\nu)$$

3.3. EM Algorithmic Convergence Criterion

The EM algorithm continues to iterate between the E-step and the M-step until some specified convergence criterion is satisfied. For example, we stop the iteration process when the change in the value of the complete data log-likelihood function $l_c(\mathbf{\Psi})$ is sufficiently small [25];

$$l_c(\mathbf{\Psi}^{(m)}) - l_c(\mathbf{\Psi}^{(m-1)}) < \epsilon \tag{3.65}$$

where $\epsilon \in (0, 1)$ is a small positive real number. In this case, we say the EM algorithm has converged and we stop the iterative process. Since the log-likelihood function is a monotonically increasing function, convergence of the EM algorithm will occur when the log-likelihood function converges to some value [13]. Alternatively, we may use the relative change in the log-likelihood as a determinant of convergence:

$$\frac{l_c(\mathbf{\Psi}^{(m)}) - l_c(\mathbf{\Psi}^{(m-1)})}{l_c(\mathbf{\Psi}^{(m)})} < \epsilon \tag{3.66}$$

3.4. Common Initialization methods for the EM Algorithm

The convergence of the EM algorithm is dependent on the selected starting values for the algorithm [6]. Whether the algorithm converges to the global maximum or a local maximum, depends on the choice of initial starting point for the EM algorithm [13]. The EM algorithm will converge to a local mode if the initial parameter values are poorly chosen [25]. In cases where the likelihood function is extremely multi-modal, there is no guarantee that the EM process will converge to the global maximum [4]. The process quiet often becomes trapped at some local maximum thereby failing to reach the global mode, especially when the process is started in the vicinity of such a local maximum. Further, if the likelihood function $L_c(\mathbf{\Psi})$ is unbounded on the edge of the parameter space $\mathbf{\Omega}$, the sequence of values $\left\{\mathbf{\Psi}^{(m)}\right\}_{m=0}^{\infty}$ generated by the EM process may diverge if the initial value of the EM algorithm (first term of the sequence) is too close to the boundary [25]. These and other problems in the EM algorithm demand that we carefully select our initial values.

In the context of finite mixtures of multivariate t-distributions, we wish to generate an optimal starting point for the EM algorithm by getting the initial value of $\mathbf{\Psi}$ denoted by $\mathbf{\Psi}^{(0)}$ and given by:

$$\mathbf{\Psi}^{(0)} = \left(\boldsymbol{\tau}^{(0)T},\ \boldsymbol{\nu}^{(0)T},\ \boldsymbol{\lambda}^{(0)T}\right)^T \tag{3.67}$$

where

$$\boldsymbol{\tau}^{(0)} = \left(\tau_1^{(0)}, \cdots, \tau_g^{(0)}\right)^T, \quad \boldsymbol{\nu}^{(0)} = \left(\nu_1^{(0)},\ \cdots,\ \nu_g^{(0)}\right)^T$$

and

$$\boldsymbol{\lambda}^{(0)} = \left(\boldsymbol{\lambda}_1^{(0)T}, \cdots, \boldsymbol{\lambda}_g^{(0)T}\right)^T \quad \text{with} \quad \boldsymbol{\lambda}_j^{(0)} = \left(\boldsymbol{\mu}_j^{(0)T}, \mathbf{\Sigma}_j^{(0)T}\right)^T$$

The vector of initial parameters $\boldsymbol{\Psi}^{(0)}$ is used as the starting point in the EM process. The vector $\boldsymbol{\Psi}^{(0)}$ is taken as the value of the model parameter $\boldsymbol{\Psi}$ at the iteration time $m = 0$. Grouping of the observed sample data $\boldsymbol{x}_o$ to obtain the initial parameter value $\boldsymbol{\Psi}^{(0)}$ occurs through initialization methods such as the k-means algorithm, hierarchical clustering and random start methods [21]. The process however, is not restricted to these methods alone [4][5][9]. A review of the k-means algorithm , hierarchical clustering and the random start methods is presented in Subsection (3.4.1), Subsection (3.4.2) and Subsection (3.4.3), respectively. The details of the burn-in scheme are presented in Section (3.4.4).

3.4.1 *k*-Means Clustering Algorithm

The k-means algorithm is used as a means of initial grouping of the observed data $\boldsymbol{x}_o$ so that a superior initial value $\boldsymbol{\Psi}^{(0)}$ is obtained for use in the EM process [11]. The k-means algorithm has the objective of partitioning the observed n data points into k clusters such that each observation $\boldsymbol{x}_i$ belongs to the cluster with the nearest mean [27]. This process results into the partitioning of the data space into Voronoi cells [32]. A Voronoi diagram is a partition of a plane into regions (cells) close to each of a given set of objects. In the simplest case, these objects are just finitely many points in the plane (called seeds, sites, or generators). The goal of k-means is to assign each data point to a cluster to such that the within cluster sum of squares is minimum [26]. In other words, k-means is a clustering method that aims to find the positions of the centers $\boldsymbol{\mu}_j^{(0)}$, for $j = 1, \cdots, k$, of the clusters that minimize the distance from the data points to the cluster centers[30].

Given an observed data vector $\boldsymbol{x}_o = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$ of dimension p, we wish to partition the n observations into $k = g$ sets (clusters), $\boldsymbol{S} = \{S_j| \text{ for } j = 1, 2, \cdots, k\}$ and $k \leq n$ so as to minimize the within-cluster sum of squares (WCSS). The goal of k-means clustering algorithm is to find the arguments to the equation

$$\arg\min_{\boldsymbol{S}} \sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in S_j} d(\boldsymbol{x}_i, \boldsymbol{\mu}_j) = \arg\min_{\boldsymbol{S}} \sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in S_j} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|_2^2 \tag{3.68}$$

where $\boldsymbol{\mu}_j$ is the mean (centroid) of cluster j, S_j is the set of all points that belong to the j^{th} cluster and $d(\boldsymbol{x}_i, \boldsymbol{\mu}_j)$ denotes the squared Euclidean distance of the point $\boldsymbol{x}_i$ to the centroid $\boldsymbol{\mu}_j$. The k-means algorithm is an iterative processes that has the following steps:

(i) *Initializing-step:* In this step, we generate the vector $\boldsymbol{\mu}_k^{(0)}$ of initial centroid values for use in the k-means algorithm;

$$\boldsymbol{\mu}_k^{(0)} = \left(\boldsymbol{\mu}_{k_1}^{(0)T}, \cdots, \boldsymbol{\mu}_{k_g}^{(0)T}\right)^T \tag{3.69}$$

where $\boldsymbol{\mu}_{k_j}^{(0)}$ for $j = 1, \cdots, g$ is the center of the j^{th} cluster. The values of the initial parameter $\boldsymbol{\mu}_k^{(0)}$, can be selected using either the Forgy

method or the random method [28]. In the Forgy method, $k = g$ observations from the data set are randomly selected and set as the initial means. The random method on the other hand, simply assigns a cluster to each of the n observations. The random method tends to cluster the means to the center of the data set while the Forgy Method spreads them out. This makes the Forgy Method an ideal initialization method for the k-means used in the EM algorithm [27][28].

(ii) *Assignment-step:* In this step, each observation $\boldsymbol{x}_i$ is assigned to one and only one cluster whose mean yields the least within-cluster sum of squares. Thus, at the $(m+1)^{th}$ iteration, each observation $\boldsymbol{x}_i$ for $i = 1, \cdots, n$, is allocated to some cluster say $S_j^{(m)}$, whose current mean $\boldsymbol{\mu}_{k_j}^{(m)}$, yields the least squared Euclidean distance $d\left(\boldsymbol{x}_i, \boldsymbol{\mu}_{k_j}^{(m)}\right)$.

$$S_j^{(m)} = \left\{\boldsymbol{x}_i : \|\boldsymbol{x}_i - \boldsymbol{\mu}_{k_j}^{(m)}\|_2^2 \leq \|\boldsymbol{x}_i - \boldsymbol{\mu}_{k_h}^{(m)}\|_2^2,\ \forall h \neq j\right\} \tag{3.70}$$

(iii) *Update-step:* At the $(m+1)^{th}$ iteration, the update step updates the centroids by computing a vector of new means,

$$\boldsymbol{\mu}_k^{(m+1)} = \left(\boldsymbol{\mu}_{k_1}^{(m+1)T}, \cdots, \boldsymbol{\mu}_{k_g}^{(m+1)T}\right)^T \tag{3.71}$$

where for $j = 1, \cdots, g$,

$$\boldsymbol{\mu}_{k_j}^{(m+1)} = \frac{1}{\left|S_j^{(m)}\right|} \sum_{\boldsymbol{x}_i \in S_j^{(m)}} \boldsymbol{x}_i \tag{3.72}$$

The k-means algorithm iterates between step *(ii)* and step *(iii)* until convergence is attained. Convergence of the k-means algorithm occurs when the assignments to the clusters no longer change. Convergence to the optimum solution is not guaranteed in this algorithm [27]. Quiet often, the process converges to a local maximum thereby yielding an inferior solution to (3.68). Like the EM algorithm, the k-means algorithm is dependent on the initial values used in the process [31].

In the context of our model (2.20) and the parameter vector (3.11), the initial value for the model parameter vector $\boldsymbol{\Psi}$ as defined in (3.67) is obtained from the final k-means algorithm clustering solution

$$\boldsymbol{S}^{(f)} = \left\{S_1^{(f)}, \cdots, S_g^{(f)}\right\} \tag{3.73}$$

which is the solution to equation (3.68). The corresponding set of the final cluster centroids is given as

$$\boldsymbol{\mu}_k^{(f)} = \left(\boldsymbol{\mu}_{k_1}^{(f)T}, \cdots, \boldsymbol{\mu}_{k_g}^{(f)T}\right)^T \tag{3.74}$$

so that the elements of the vector (3.67), for $j = 1, \cdots, g$, are determined as follows:

$$\boldsymbol{\mu}_j^{(0)} = \boldsymbol{\mu}_{k_j}^{(f)}, \qquad \boldsymbol{\Sigma}_j^{(0)} = \frac{1}{\left|S_j^{(f)}\right|} \sum_{\boldsymbol{x}_i \in S_j^{(f)}} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_{k_j}^{(f)}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_{k_j}^{(f)}\right)^T$$

$$\tau_j^{(0)} = \frac{\left|S_j^{(f)}\right|}{n} \quad \text{and degrees of freedom are set at} \quad \nu_j^{(o)} = 4$$

These generated set of initial values from the the k-means algorithm are used as an optimized starting point for the EM algorithm [21].

3.4.2 Hierarchical Clustering Algorithm

Hierarchical clustering is a method of cluster analysis which seeks to partition the observed data $\boldsymbol{x}_o$ into g sets (clusters) by building a hierarchy of clusters. Two general methods used to achieve hierarchical clustering are the agglomerative and the divisive methods [34]. In the agglomerative approach, each observation starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy. This continues until an optimal solution is achieved. In the divisive method, all observations start in one cluster, which is then split recursively as one moves down the hierarchy. In hierarchical cluster analysis, an appropriate metric and a linkage criterion which specifies the dissimilarity of clusters as a function of the pairwise distances of observations in the clusters, must be used as a measure of the dissimilarity between clusters. Several agglomerative methods are used in hierarchical clustering. The average-linkage (ALINK), single-linkage (SLINK) and the complete-linkage (CLINK) are some of the agglomerative algorithms used in the hierarchical clustering [33]. In the CLINK hierarchical clustering, the metric commonly used is the squared Euclidean distance and the linkage criterion is that of maximum distance [33]. The EM algorithm uses the CLINK approach when initializing the model parameter [10].

Given an observed data vector $\boldsymbol{x}_o = \left(\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T\right)^T$ of dimension p, we wish to partition the n observations into g sets, $\boldsymbol{S} = \{S_j| \quad \text{for} \quad j = 1, 2, \cdots, g\}$ and $g \leq n$, so as to minimize the inter-cluster squared Euclidean distance defined using the maximum-linkage criterion;

$$D(S_h, S_k) = \max_{\boldsymbol{x}_h \in S_h,\ \boldsymbol{x}_k \in S_k} d(\boldsymbol{x}_h,\ \boldsymbol{x}_k) \tag{3.75}$$

where d is the chosen metric, in this case the squared Euclidean distance, $\boldsymbol{x}_k$ and $\boldsymbol{x}_h$ are data points belonging to sets (clusters) S_k and S_h, respectively, $h, k \in \{1, \cdots, g\}$. The following summarizes the CLINK algorithm as applied in the initialization of the EM algorithm;

(i) Assign a cluster, $S_i^{(0)}$ to each of the observed data points $\boldsymbol{x}_i$, for $i = 1, 2, \cdots, n$ so that we have n clusters.

$$\boldsymbol{S}^{(0)} = \{S_i^{(0)}|\ \boldsymbol{x}_i \in S_i^{(0)} \quad \text{for} \quad i = 1, \cdots, n\}$$

where each of the observation is considered a prototype of the respective cluster to which it belongs.

(ii) Using the distance function (3.75), determine the inter-cluster distances for all the groups.

(iii) Merge the two clusters that have the minimum inter-cluster distance to form a new cluster. The newly formed cluster is taken to be a single cluster. Thus, the total number of clusters reduces to $n-1$.

(iv) Repeat steps (ii) and (iii). Stop the process when exactly g clusters remain;

$$\boldsymbol{S}^{(f)} = \left\{ S_1^{(f)}, \cdots, S_g^{(f)} \right\}$$

The partition $\boldsymbol{S}^{(f)}$ is taken to be the optimized grouping of the data set $\boldsymbol{x}_o$ into g groups or clusters. The formed g clusters are then used as an initial partition of the data set for the computation of the initial vector of parameters (3.67) for use in the full EM algorithm. In the context of our model (2.20) and the parameter vector (3.11), the initial value for the model parameter vector $\boldsymbol{\Psi}$ as defined in (3.67) is obtained using $\boldsymbol{S}^{(f)}$ as follows:

$$\boldsymbol{\mu}_j^{(0)} = \sum_{\boldsymbol{x}_i \in S_j^{(f)}} \boldsymbol{x}_i, \qquad \boldsymbol{\Sigma}_j^{(0)} = \frac{1}{\left|S_j^{(f)}\right|} \sum_{\boldsymbol{x}_i \in S_j^{(f)}} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(0)}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(0)}\right)^T$$

$$\tau_j^{(o)} = \frac{\left|S_j^{(f)}\right|}{n} \quad \text{and degrees of freedom are set at} \quad \nu_j^{(0)} = 4$$

for all $j = 1, \cdots, g$. Note again that the degrees of freedom for all g-components are initially set as $\nu_j = 4$ [21].

3.4.3 Random Start Methods

One way of specifying an initial partition of the data is to randomly divide the data into g groups corresponding to the g components of the mixture model [35]. With this approach, the observed data vector $\boldsymbol{x}_o$ is randomly partitioned into g-groups, say $\boldsymbol{S} = \{S_1, \cdots, S_g\}$. Using the groups (clusters) $S_1, \cdots, S_g$ as the sub-samples, a vector of statistics computed using these sample data can be set as the vector of initial parameter values for implementation in the EM algorithm. Thus, the first M-step of the EM process is performed on the basis of the set sub-samples, provided the sample is large enough [35]. This is a simple method commonly used to initialize the EM algorithm (see [1], [2] and [25]). With random starts, the effect of the central limit theorem tends to have the component parameters initially being similar at least when the underlying sample is large [25]. Alternatively, in the context of our model (2.20), the parameter vector (3.11) and the complete data vector (3.22), the initial parameter vector (3.67)

The EM algorithm can be initialized at random as follows:

(i) Calculate $\overline{\boldsymbol{x}}$ and $\boldsymbol{S^2}$ using the observed sample data $\boldsymbol{x}_o$;

$$\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \quad \text{and} \quad \boldsymbol{S^2} = \frac{1}{n}\sum_{i=1}^{n} \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T$$

(ii) Randomly generate a set of g values, $\boldsymbol{\mu}_1^{(0)}, \cdots, \boldsymbol{\mu}_g^{(0)}$, independently from the multivariate t-distribution, $t_p(\overline{\boldsymbol{x}}, \boldsymbol{S^2}, 4)$ as

$$\boldsymbol{\mu}_1^{(0)}, \cdots, \boldsymbol{\mu}_g^{(0)} \quad \widetilde{\text{i.i.d}} \quad t_p(\overline{\boldsymbol{x}}, \boldsymbol{S^2}, 4)$$

(iii) The initial component covariance matrices, contained in the vector $\boldsymbol{\Sigma}^{(0)} = \left(\boldsymbol{\Sigma}_1^{(0)}, \cdots, \boldsymbol{\Sigma}_g^{(0)}\right)^T$, are specified as

$$\boldsymbol{\Sigma}_j^{(0)} = \boldsymbol{S^2} \qquad \text{for all} \quad j = 1, \cdots, g$$

(iv) The mixing proportions for the components contained in the vector $\boldsymbol{\tau}^{(0)} = \left(\tau_1^{(0)}, \cdots, \tau_g^{(0)}\right)^T$ are specified as

$$\tau_j^{(0)} = \frac{1}{g} \qquad \text{for all} \quad j = 1, \cdots, g$$

(v) The value for the degrees of freedom parameter is preset to a value of $\boldsymbol{\nu}^{(0)} = (4, 4, \cdots, 4)^T$. Each component has degrees of freedom set at $\nu = 4$ (see [21])

This method has the advantage of spreading out the initial component means $\boldsymbol{\mu}_j^{(0)}$ as compared to the method of just randomly grouping data into g classes.

3.4.4 Burn-in Scheme

In an attempt to improve the performance of the EM algorithm as a tool for maximum likelihood estimation in mixture models featuring Gaussian components, [4] developed the burn-in scheme as an alternative initialization procedure for the model parameters to methods that employ random start, k-means algorithm or hierarchical clustering. In their method, [4] combine the techniques already available in other adaptations to the general EM algorithm such as the Expectation Conditional Maximization (ECM) algorithm [15], the emEM algorithm [5], the Multi-cycle Expectation Maximization algorithm [1] and the Sparse Expectation Maximization (SEM) algorithm [9]. The performance of the burn-in scheme as compared to the hierarchical clustering method is well documented in [4], where the mixtures used feature Gaussian Components. The techniques of the burn-in scheme have been suggested to be a promising extension to mixture models featuring non-Gaussian models [4]. This section presents the extension of the techniques of this scheme to mixture models featuring t-components.

In the context of the model (2.20), the parameter vector (3.11) and the complete data vector (3.22), the initial parameter vector (3.67) is computed using the optimally generated value of the $\boldsymbol{Z}$ matrix from the following scheme:

(i) Generate a finite set of $n \times p$ matrices whose entries are the component indicator variable values, i.e generate the set

$$\mathcal{A}_z = \left\{ \boldsymbol{z}_1^{(0)}, \cdots, \boldsymbol{z}_{2^b}^{(0)} \right\} \qquad \text{for some } b \in \mathbb{N}$$

where $\boldsymbol{z}_a^{(0)} \in \mathcal{A}_z$ denotes a unique partition of the data vector $\boldsymbol{x}_o$ into g groups, $\boldsymbol{S}_a = \left\{ S_{a_1}, S_{a_2} \cdots, S_{a_g} \right\}$, with entries $z_{ij} = 1$ if observation $\boldsymbol{x}_i \in S_{a_j}$ and $z_{ij} = 0$ otherwise, for $a = 1, \cdots, 2^b$.

(ii) For each element $\boldsymbol{z}_a^{(0)} \in \mathcal{A}_z$, i.e. for all $a = 1, \cdots, 2^b$, compute the values of the vector of initial parameters

$$\boldsymbol{\Psi}_a^{(0)} = \left(\boldsymbol{\tau}_a^{T^{(0)}}, \boldsymbol{\nu}_a^{T^{(0)}}, \boldsymbol{\lambda}_a^{T^{(0)}} \right)^T \tag{3.76}$$

where;

$$\boldsymbol{\tau}_a^{(0)} = \left(\tau_{a_1}^{(0)}, \cdots, \tau_{a_g}^{(0)} \right)^T, \qquad \boldsymbol{\nu}_a^{(0)} = (4, \cdots, 4)^T$$

and

$$\boldsymbol{\lambda}_a^{(0)} = \left(\left(\boldsymbol{\mu}_{a_1}^{T^{(0)}}, \boldsymbol{\Sigma}_{a_1}^{T^{(0)}} \right)^T, \cdots, \left(\boldsymbol{\mu}_{a_g}^{T^{(0)}}, \boldsymbol{\Sigma}_{a_g}^{T^{(0)}} \right)^T \right)^T$$

.

Note that the initial values for the degrees of freedom parameter in our inference will be preset to a value of 4 for each t-component of the mixture model [21]. Thus, we have $\nu_j = 4$ for all $j = 1, \cdots, g$. Further,

$$\tau_{a_j}^{(0)} = \frac{\left| S_{a_j} \right|}{n} \qquad \forall j = 1, \cdots, g \tag{3.77}$$

,

$$\boldsymbol{\mu}_{a_j}^{(0)} = \frac{1}{\left| S_{a_j} \right|} \sum_{\boldsymbol{x}_i \in S_{a_j}} \boldsymbol{x}_i \qquad \forall j = 1, \cdots, g \tag{3.78}$$

and

$$\boldsymbol{\Sigma}_{a_j}^{(0)} = \frac{1}{\left| S_{a_j} \right|} \sum_{\boldsymbol{x}_i \in S_{a_j}} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_{a_j}^{(0)} \right) \left(\boldsymbol{x}_i - \boldsymbol{\mu}_{a_j}^{(0)} \right)^T \quad \forall j = 1, \cdots, g \tag{3.79}$$

This step generates a set of 2^b possible best initial parameter values

$$\mathcal{A}_{\boldsymbol{\Psi}} = \left\{ \boldsymbol{\Psi}_a^{(0)} |\ a = 1, \cdots, 2^b \right\}$$

where $\boldsymbol{\Psi}_a^{(0)}$ is generated using $\boldsymbol{z}_a^{(0)}$. The actual best initial parameter value for use in the full EM process is denoted by $\boldsymbol{\Psi}_a^{(f)}$ and is the emerging vector from the burn-in scheme.

(iii) For each matrix $\boldsymbol{z}_a^{(0)} \in \mathcal{A}_z$, conduct a single pair of the EM algorithm steps using the corresponding $\boldsymbol{\Psi}_a^{(0)} \in \mathcal{A}_{\boldsymbol{\Psi}}$ as the vector of initial parameter values. In the context of mixtures of t-distributions, the E-step and the M-step are carried out according to the description in

section (3.2.1) and (3.2.2). Taking $m = 0$ on the first iteration , the E-step is effected by computing;

$$e_{a_{ij}}^{(0)} = \frac{\tau_{a_j}^{(0)} f\left(\boldsymbol{x}_i | \boldsymbol{\mu}_{a_j}^{(0)},\ \boldsymbol{\Sigma}_{a_j}^{(0)}\right)}{\sum_{k=1}^{g} \tau_{a_k}^{(0)} f\left(\boldsymbol{x}_i | \boldsymbol{\mu}_{a_k}^{(0)},\ \boldsymbol{\Sigma}_{a_k}^{(0)}\right)} \tag{3.80}$$

$$\sigma_{a_{ij}}^{(0)} = \frac{\nu_{a_j}^{(0)} + p}{\nu_{a_j}^{(0)} + \left(\boldsymbol{x}_i - \boldsymbol{\mu}_{a_j}^{(0)}\right)^T \left(\boldsymbol{\Sigma}_{a_j}^{(0)}\right)^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_{a_j}^{(0)}\right)} \tag{3.81}$$

and

$$\xi_{a_{ij}}^{(0)} = \ln \sigma_{a_{ij}}^{(0)} + \psi\left(\frac{\nu_{a_j}^{(0)} + p}{2}\right) - \ln\left(\frac{\nu_{a_j}^{(0)} + p}{2}\right) \tag{3.82}$$

On the first M-step, the parameter vector (3.76), is updated to obtain the vector

$$\boldsymbol{\Psi}_a^{(1)} = \left(\boldsymbol{\tau}_a^{T^{(1)}}, \boldsymbol{\nu}_a^{T^{(1)}}, \boldsymbol{\lambda}_a^{T^{(1)}}\right)^T \tag{3.83}$$

using;

$$\tau_{a_j}^{(1)} = \frac{1}{n} \sum_{i=1}^{n} e_{a_{ij}}^{(0)} \tag{3.84}$$

$$\boldsymbol{\mu}_{a_j}^{(1)} = \frac{\sum_{i=1}^{n} e_{a_{ij}}^{(0)} \sigma_{a_{ij}}^{(0)} \boldsymbol{x_i}}{\sum_{i=1}^{n} e_{a_{ij}}^{(0)} \sigma_{a_{ij}}^{(0)}} \tag{3.85}$$

and

$$\boldsymbol{\Sigma}_{a_j}^{(1)} = \frac{\sum_{i=1}^{n} e_{a_{ij}}^{(0)} \sigma_{a_{ij}}^{(0)} \left(\boldsymbol{x_i} - \boldsymbol{\mu}_{a_j}^{(1)}\right) \left(\boldsymbol{x_i} - \boldsymbol{\mu}_{a_j}^{(1)}\right)^T}{\sum_{i=1}^{n} e_{a_{ij}}^{(0)} \sigma_{aij}^{(0)}} \tag{3.86}$$

for all $a = 1, \cdots, 2b$ and $j = 1, \cdots, g$

(iv) Evaluate the respective log-likelihood values $\ell_a^{(1)}$, corresponding to each of the vectors $\boldsymbol{\Psi}_a^{(0)}$ for all $a = 1, \cdots, 2^b$. This step generates a set of the log-likelihoods, evaluated at the first iteration.

$$\mathcal{A}_l = \left\{\ell_1^{(1)}, \cdots, \ell_{2^b}^{(1)}\right\} \tag{3.87}$$

using

$$\ell_a^{(1)} = l_c\left(\boldsymbol{\Psi}_a^{(1)}\right) \quad \text{where} \quad l_c(.) \text{ is defined according to (3.26)}$$

(v) Rank the elements of the set $\mathcal{A}_l$ from the largest to the smallest value

$$\ell_1'^{(1)} \geq \ell_2'^{(1)} \geq \cdots \geq \ell_{(2^b-1)}'^{(1)} \geq \ell_{2^b}'^{(1)} \tag{3.88}$$

(vi) Rank the elements of the set $\mathcal{A}_{\boldsymbol{\Psi}}$ according to the rankings of their respective log-likelihoods in (3.88)

$$\boldsymbol{\Psi}_1^{\prime(0)} \geq \boldsymbol{\Psi}_2^{\prime(0)}, \geq \cdots, \geq \boldsymbol{\Psi}_{(2^b-1)}^{\prime(0)} \geq \boldsymbol{\Psi}_{2^b}^{\prime(0)} \tag{3.89}$$

(vii) Rank the elements of the set $\mathcal{A}_z$ according to the rankings of their corresponding $\boldsymbol{Z}^{(0)}$ matrices in (3.89)

$$\boldsymbol{z}_1^{\prime(0)} \geq \boldsymbol{z}_2^{\prime(0)}, \geq \cdots, \geq \boldsymbol{z}_{(2^b-1)}^{\prime(0)} \geq \boldsymbol{z}_{2^b}^{\prime(0)} \tag{3.90}$$

(viii) From the ranked elements in (3.90), discard (burn off) the lower half of the values. This discards half the possible values of the $\boldsymbol{Z}$ matrix that are more likely to produce inferior starting points.

(ix) For the remaining upper half elements of (3.90), repeat steps (iii)–(viii) until only one value of the $\boldsymbol{Z}$ matrix remains, say $\boldsymbol{z}_f^{(0)}$.

$\boldsymbol{z}_f^{(0)}$ is an optimized values of the $\boldsymbol{Z}$ matrix. Implement the full EM algorithm using $\boldsymbol{\Psi}_f^{(0)}$ obtained from $\boldsymbol{z}_f^{(0)}$, as an optimized EM starting point.

CHAPTER 4
Computational Results and Analysis

This chapter presents a summary of the experimental results of this study. Section 4.1 outlines the strategies that are employed in the computations and Section 4.2 presents the analysis strategy. Section 4.3 and Section 4.4 present the results and analysis from the computations.

4.1. Computational Strategies in R

To initiate the EM algorithm through random start, k-means algorithm or hierarchical clustering, a number of packages are available for use in the statistical software **R** [44]. In this study, we use the function `init.mix` in the package `EMMIXskew` (see [21] for details). For the burn-in scheme, we write the required code in **R**, making use of the function `initEmmix` in the package `EMMIXskew`.

4.1.1 k-means algorithm in R

Initialization via k-means can be started either from one partition or from several partitions of $\boldsymbol{x}_o$. Let $1k$-means denote the initialization where only one trial of k-means is used to obtain $\boldsymbol{\Psi}^{(0)}$ and let $10k$-means be the initialization where ten different trials of the k-means are selected and the k-means producing the superior log-likelihood is chosen for use in the full EM algorithm. For example, taking the data set to be the `iris`:

```
> iris.Fit1.init.1 <- init.mix(dat=iris[,1:4],g=3,distr="mvt",
+ ncov=2,nkmeans=1,nrandom=0,nhclust=FALSE,maxloop=10)
> iris.Fit1.init.2 <- init.mix(dat=iris[,1:4],g=3,distr="mvt",
+ ncov=2, nkmeans=10,nrandom=0,nhclust=FALSE,maxloop=1)
```

which generates two sets of $\boldsymbol{\Psi}^{(0)}$ stored in the **R** objects `iris.Fit1.init.1` and `iris.Fit1.init.2`. The object `iris.Fit1.init.1` contains initial parameter values generated using only a single k-means partition while the object `iris.Fit1.init.2` is obtained using the superior of the ten k-means partitions.

4.1.2 Random start method in R

When a single starting point is randomly selected and the full EM is run from this point, we will call this initialization procedure 1Random (1Rand in short). It may be ideal to obtain several partitions, for example 10, run a preset number of EM iterations on each partition and select the superior partition for use in the full EM algorithm. We call this initialization 10Random (10Rand in short). For example, for the data set `dat2`, 1Rand and 10Rand maybe executed respectively, as

```
> dat2.Fit2.init.1 <- init.mix(dat=dat2, g=2, distr="mvt",
+ ncov=2, nkmeans=0, nrandom=1, nhclust=FALSE, maxloop=10)

> dat2.Fit2.init.2 <- init.mix(dat=dat2, g=2, distr="mvt",
+ ncov=2, nkmeans=0, nrandom=10, nhclust=FALSE,maxloop=1)
```

The object `dat2.Fit2.init.1` contains the initial parameters generated using a random method with a single partition of the data. The object `dat2.Fit2.init.2` contains the initial parameter values generated using 10 different partitions of the data, and selecting the best partition.

4.1.3 Hierarchical Clustering in R

When generating the initial parameter values using hierarchical clustering, the maximum number of iterations is not pre-set but attained when the number of clusters attained is g. For example, taking the data set to be `banknote`, we obtain initial values stored in the object `bank.Fit3.init`, using the command:

```
> bank.Fit3.init <- init.mix(banknote[,2:7], g=2, distr="mvt",
+ ncov=2,nkmeans=0, nrandom=0, nhclust=TRUE)
```

4.1.4 Burn-in concepts using R

In this study, the initial candidate $\boldsymbol{Z}$ matrices are selected through data clustering. The observed data set $\boldsymbol{x}_o$ is partitioned into g clusters so that each data point $\boldsymbol{x}_i$ for $i = 1, 2, \cdots, n$ belongs to only one of the g clusters. For each clustering, their corresponds only one initial value of the $\boldsymbol{Z}$ matrix (see Subsection 3.4.4 of this dissertation). Thus, selecting 2^b possible clusters of the observed data set $\boldsymbol{x}_o$ corresponds to selecting 2^b possible initial $\boldsymbol{Z}$ matrices. In this study, we select the $\boldsymbol{Z}$ matrices through selecting the corresponding clusters. The **R** function `initEmmix` then is used to generate the required vector of initial parameters $\boldsymbol{\Psi}^{(0)}$ via the **R** command:

```
> bank.optimal_Z <- bankclust1
> bank.optimal_init <- initEmmix(banknote[,2:7], g=2,
+ clust=bank.optimal_Z,distr="mvt", ncov=2, maxloop=10)
```

where `bank.optimal_Z` is the best $\boldsymbol{Z}$ matrix from the burn-in scheme, when the `banknote` data set is used. The generated set of initial values are stored in the list structure `bank.optimal_init`.

For a detailed example on the burn-in scheme and the associated **R** code the reader is referred to the Appendix of this dissertation.

The functions `init.mix` and `initEmmix` of the package `EMMIXskew` in **R** return an array structure with the values:

(i) `pro`, a numeric vector of the component mixing probabilities with the j^{th} entry τ_j corresponding to the mixing probability for the j^{th} component.

(ii) `mu`, a $p \times g$ matrix with the j^{th} column as the corresponding mean for the j^{th} component of the mixture model.

(iii) `sigma`, a three dimensional $p \times p \times g$ array with its j^{th} component matrix (p, p, j) as the covariance matrix for the j^{th} component of the mixture model.

(iv) `dof`, a vector of degrees of freedom for each component with the j^{th} entry ν_j corresponding to the degrees of freedom for the j^{th} component.

In the context of our model (2.20), the parameter vector (3.11) and the initial parameter vector (3.67), the output from EM initialization in **R** is summarized in Table 4.1. From Table 4.1, note that the set of initial values of the mixture model parameter $\boldsymbol{\Psi}$ can be written as

$$\boldsymbol{\Psi}^{(0)} = \left(\boldsymbol{\tau}^{(0)^T}, \boldsymbol{\nu}^{(0)^T}, \boldsymbol{\lambda}^{(0)^T}\right)^T = \left(\boldsymbol{\tau}^{(0)^T}, \boldsymbol{\nu}^{(0)^T}, \boldsymbol{\mu}^{(0)^T}, \boldsymbol{\Sigma}^{(0)^T}\right)^T \tag{4.1}$$

where $\boldsymbol{\Psi}^{(0)}$ is understood to be the output from either the function `init.mix` or `initEmmix` of the package `EMMIXskew` in **R**.

Parameter	**R** argument	Dimensions	Description
$\boldsymbol{\tau}^{(0)}$	`pro`	$g \times 1$	Component Mixing Probabilities
$\boldsymbol{\nu}^{(0)}$	`dof`	$g \times 1$	Degrees of Freedom
$\boldsymbol{\mu}^{(0)}$	`mu`	$p \times g$	Location parameter
$\boldsymbol{\Sigma}^{(0)}$	`sigma`	$p \times p \times g$	Scale Matrix

Table 4.1: Structure of the initial parameter value $\boldsymbol{\Psi}^{(0)}$ for mixtures of multivariate t-distributions in EMMIXskew.

4.2. Analytical Strategies in R

To implement the full EM algorithm in **R**, we use the functions `EmSkewfit1` and `EmSkewfit2` in the package `EMMIXskew` (see [21] for details). In this study, to compare the performance of the EM algorithm initialized using the four methods, the following EM output values are analysed:

(a) **Convergent log-likelihood value $l(\hat{\boldsymbol{\Psi}})$:** This is the value of the log-likelihood function $l(\boldsymbol{\Psi})$ at the point of convergence of the EM algorithm [2][13]. The convergent value of the log-likelihood function is the main feature used to asses the performance of each initialization method [5]. The initialization method that yields the largest value of $l(\hat{\boldsymbol{\Psi}})$ is considered to be the optimal performing method [4].

Measuring the effectiveness of an initialization method using the maximization of the log-likelihood is not always an effective method as the log-likelihood function may be unbounded. To ensure that the likelihood function does not diverge, the choice for the covariance matrices must be one that ensures the same determinants for all component covariances [5]. In our experiment, we restrict the covariance matrices to a common diagonal variance [21].

(b) **Akaike Information Criterion (AIC):** Suppose that we have a mixture model of some given data. Let k be the total number of estimated parameters in the model. If $l(\hat{\boldsymbol{\Psi}})$ is the value of the log-likelihood function at the point of convergence of the EM algorithm, then the Akaike information criterion (AIC) value for the model is given by

$$\text{AIC} \;=\; 2k - 2l(\hat{\boldsymbol{\Psi}})$$

The Akaike information criterion is a criterion for model selection among a finite set of models [21]. When fitting data to finite mixtures models, it is possible to increase the likelihood function by adding more parameters [4]. This may result in over-fitting. The AIC resolves this problem by introducing a penalty term for the number of parameters in the model [43]. The AIC is used as an estimator of the relative quality of statistical models for a given data set [4]. In this study, we assume that each initialization method gives a statistical model different from the other methods. The AIC analyzes the quality of each initialization method by estimating the quality of the corresponding resulting model. Given a set of candidate models, the best model is one with the minimum AIC value [4][12].

(c) **Bayes Information Criterion (BIC):** Like the Akaike information criterion, the Bayes information criterion is also used in model selection when we have a number of possible models [43]. Formally, BIC is defined as

$$\text{BIC} \;=\; k\ln(n) - 2l(\hat{\boldsymbol{\Psi}})$$

where n is the size of the fitted data, k is the total number of estimated parameters in the model and $l(\hat{\boldsymbol{\Psi}})$ is the value of the log-likelihood

function at the point of convergence of the EM algorithm. The preferred model is the one with the minimum BIC value [4][43]. Thus, we will choose the best initialization method based on the BIC value in the corresponding resulting model.

(d) **Convergence Error:** The EM algorithm either converges within the given number of iteration or fails to converge. If the algorithm converges at or before the preset maximum number of iterations (`itmax`), then the convergence error has value 0, if the algorithm did not converge within the preset `itmax` number of iterations, then the convergence error has value 1. If the EM algorithm converges to a point of singularity, then the convergence error has value 2 [11].

(e) **Error Rate:** The component from which any particular observed data point $\boldsymbol{x}_i$ originates is known and is indicated by a categorical variable or lable [11]. For example, the `iris` data has the categorical variable `Species`, which will indicate whether a particular point is from component one (the `setosa` group), component two (the `versicolor` group) or component three (the `virginica` group). By plotting the contours of the fitted model, we can compare the final clustering of individual data points in our model to the true grouping in the data if it is known [12]. The number of miss-allocated data points in a given model measures the error rate for the given model [11][12]. Thus, error rate can also be used to determine the quality of the model and hence, the performance of the employed initialization method [11].

For more details on all the output values from implementing the EM algorithm in the statistical software **R**, the reader is referred to [4], [10], [11], [12] and [21].

4.3. Computational Results from Simulated Data Sets

This section presents a summary of results of fitting four simulated data sets to finite mixtures of multivariate t-distributions. We initialize the EM algorithm using burn-in concepts, k-means clustering algorithm, random start and hierarchical clustering. The simulated data sets are summarized in Table 4.2. Note that the component distributions (comp.distr) of the simulated data sets are either t-distributions or normal distributions.

Data	Dimension (p)	Groups (g)	Sample size (n)	comp.distr
`dat1`	2	3	1000	t
`dat2`	3	4	800	t
`dat3`	2	3	1000	normal
`dat4`	3	4	800	normal

Table 4.2: A summary of the four simulated illustrative data sets.

4.3.1 Simulated bivariate data from a 3-component mixture of t-distributions.

We generate a $p = 2$ dimensional sample data (`dat1`) from a mixture of t-distributions featuring three components whose respective parameters are;

$$\boldsymbol{\mu}_1 = (75.5, 1.85)^T, \qquad \boldsymbol{\mu}_2 = (62.5, 1.55)^T, \qquad \boldsymbol{\mu}_3 = (82.5, 1.83)^T,$$

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 193.92 & 1.0605 \\ 1.0605 & 0.0095 \end{pmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 119.15 & 0.6376 \\ 0.6376 & 0.0068 \end{pmatrix},$$

$$\boldsymbol{\Sigma}_3 = \begin{pmatrix} 153.92 & 0.6535 \\ 0.6535 & 0.0062 \end{pmatrix},$$

$$\boldsymbol{\tau} = (0.35,\ 0.35,\ 0.30)^T \qquad \text{and} \qquad \boldsymbol{\nu} = (5, 5, 5)^T$$

The simulated data set 1 (which will be denoted `dat1` in **R**), is shown in the scatter plot of Figure 4.1. The distribution of the respective variables are shown in Figure 4.2. Fitting the simulated data set `dat1` to a 3-component mixture of multivariate t-distributions via the EM algorithm gives the averages presented in Table 4.3 from the 142 simulations. The EM algorithm is initialized using (a) k-means clustering algorithm, (b) random start method, (c) hierarchical clustering and (d) burn-in scheme.

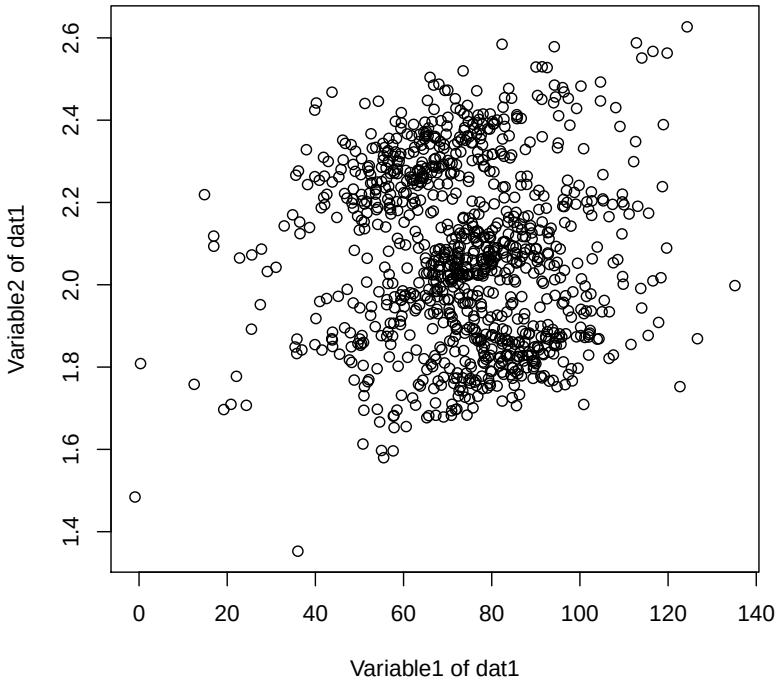

Figure 4.1: Scatter plot for 1000 data points in the simulated data (dat1)

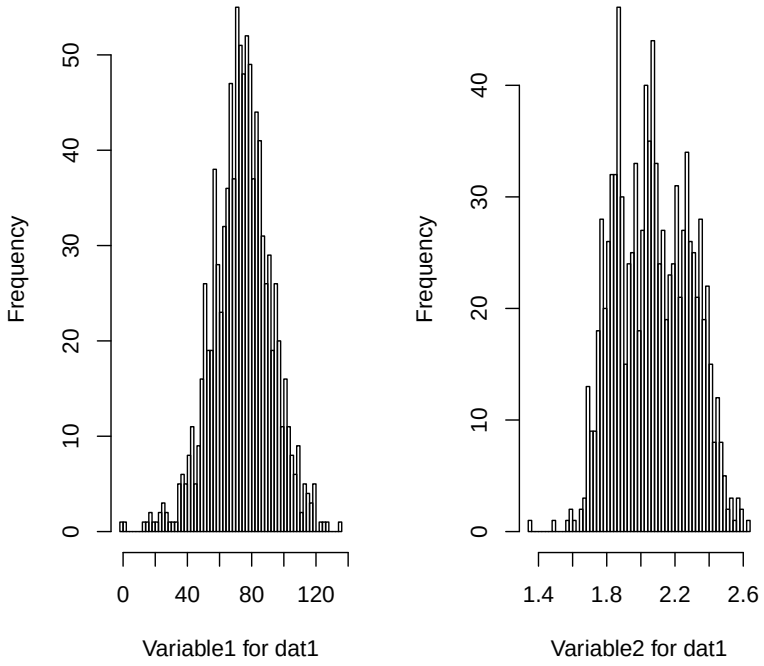

Figure 4.2: Distribution of the two variables in the simulated data (dat1)

	k-means	Random start	Hierarchical	Burn-in
`conv.error`	0	0	0	0
`BIC`	8405.261	8323.321	8503.622	8323.022
`AIC`	8339.461	8259.520	8439.820	8259.221
$l(\hat{\mathbf{\Psi}})$	-4156.73	-4116.760	-4206.910	-4116.610

Table 4.3: Summary of the average EM output values from 142 simulations of fitting dat1 to mixtures of 3 bivariate t-distributions via EM algorithm initialized using (a) k-means clustering algorithm, (b) random start method, (c) hierarchical clustering and (d) burn-in scheme.

Comparing the AIC and BIC values from Table 4.3, we note that the model corresponding to the burn-in scheme produced the least values of 8259.221 and 8323.022 as AIC and BIC values, respectively. Therefore, we concluded that for the simulated data set `dat1`, the burn-in scheme outperformed both the k-means algorithm and the hierarchical clustering as methods of initializing model parameters for the EM algorithm. Further, we see from Table 4.3 that the convergence error for all the EM algorithms was coded 0, indicating that the algorithms converged within the set number of maximum iterations of `itmax=1000`.

For the simulated data set `dat1`, the optimal model fitted was obtained

from the burn-in initialization method. The average maximum convergent log-likelihood based on 142 trials was determined to be $l(\hat{\boldsymbol{\Psi}}) = -4116.610$. The least preferred model with average convergent log-likelihood of $l(\hat{\boldsymbol{\Psi}}) = -4206.910$ was obtained from the hierarchical clustering method.

The distributions of the convergent log-likelihood values from the four initialization methods are given in Figure 4.3. The four initialization methods include the k-means clustering algorithm, the hierarchical clustering algorithm, the random start method and the burn-in scheme. To compare the performance of the burn-in scheme with those of the other methods, the average convergent log-likelihood from the EM initialized via burn-in scheme is included as a purple vertical line on each graph. Figure 4.3 suggests that EM initializations via the burn-in scheme leads to a higher log-likelihood value in a higher percentage of cases than the other methods. Further, note that initializations via hierarchical clustering results in a uniform convergent log-likelihood of $l(\hat{\boldsymbol{\Psi}}) = -4206.910$ for all the 142 trials. Based on the distributions of the convergent log-likelihoods in Figure 4.3, we conclude that the burn-in scheme is a superior initialization method of the four methods.

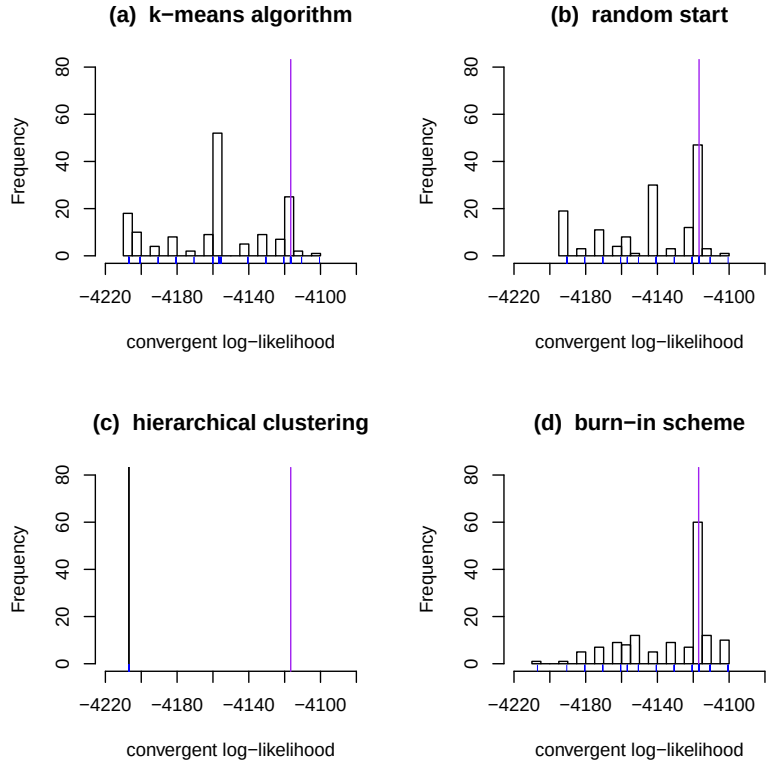

Figure 4.3: Distribution of Convergent log-likelihoods for the simulated data dat1 fitted to a 3-component mixture of t-distributions via the EM algorithm initialized using (a) k-means (b) random starts (c) hierarchical clustering and (d) burn-in.

Table 4.4 and Table 4.5 present a comparison of the true parameter values with the estimates from the EM algorithm initialized via different methods

for the simulated data set `dat1`. From the two tables, we can see that the estimates of the parameter from the EM algorithm initialized via the burn-in scheme are closer to the true population values.

Par.	True Par. Value	Par. Estimate When EM is Initialized via (a) Random Start	(b) Burn-in Scheme
$\hat{\boldsymbol{\tau}}$	$(.35, .35, .30)^T$	$(.32, .31, .37)^T$	$(.31, .31, .38)^T$
$\hat{\boldsymbol{\nu}}$	$(5, 5, 5)^T$	$(5.73, 6.50, 3.91)^T$	$(5.62, 7.10, 3.54)^T$
$\hat{\boldsymbol{\mu}}_1$	$(75.5, 1.85)^T$	$(68.311, 2.312)^T$	$(76.421, 2.061)^T$
$\hat{\boldsymbol{\mu}}_2$	$(62.5, 1.55)^T$	$(76.445, 2.061)^T$	$(68.244, 2.312)^T$
$\hat{\boldsymbol{\mu}}_3$	$(82.5, 1.83)^T$	$(77.381, 1.836)^T$	$(77.406, 1.836)^T$
$\hat{\boldsymbol{\Sigma}}_1$	$\begin{pmatrix} 193.92 & \\ 1.0605 & .0095 \end{pmatrix}$	$\begin{pmatrix} 206.371 & \\ 0 & .0066 \end{pmatrix}$	$\begin{pmatrix} 203.276 & \\ 0 & .0065 \end{pmatrix}$
$\hat{\boldsymbol{\Sigma}}_2$	$\begin{pmatrix} 119.15 & \\ .6376 & .0068 \end{pmatrix}$	$\begin{pmatrix} 206.371 & \\ 0 & .0066 \end{pmatrix}$	$\begin{pmatrix} 203.276 & \\ 0 & .0065 \end{pmatrix}$
$\hat{\boldsymbol{\Sigma}}_3$	$\begin{pmatrix} 153.92 & \\ .6535 & .0062 \end{pmatrix}$	$\begin{pmatrix} 206.371 & \\ 0 & .0066 \end{pmatrix}$	$\begin{pmatrix} 203.276 & \\ 0 & .0065 \end{pmatrix}$

Table 4.4: Average EM algorithm output values from the 142 simulations of fitting dat1 to 3-component mixtures of t-distributions, EM initialized via (a) random start and (b) burn-in scheme.

Par.	True Par. Value	Par. Estimate When EM is Initialized via (c) k-means Algorithm	(d) Hierarchical
$\hat{\boldsymbol{\tau}}$	$(.35, .35, .30)^T$	$(.35, .57, .08)^T$	$(.01, .01, .98)^T$
$\hat{\boldsymbol{\nu}}$	$(5, 5, 5)^T$	$(7.38, 18.55, 29.50)^T$	$(24.97, 4.00, 21.29)^T$
$\hat{\boldsymbol{\mu}}_1$	$(75.5, 1.85)^T$	$(66.2457, 2.2514)^T$	$(73.9321, 2.0712)^T$
$\hat{\boldsymbol{\mu}}_2$	$(62.5, 1.55)^T$	$(76.8463, 1.9282)^T$	$(0.0000, 0.0000)^T$
$\hat{\boldsymbol{\mu}}_3$	$(82.5, 1.83)^T$	$(85.6455, 2.3161)^T$	$(81.4655, 1.9740)^T$
$\hat{\boldsymbol{\Sigma}}_1$	$\begin{pmatrix} 193.92 & \\ 1.0605 & .0095 \end{pmatrix}$	$\begin{pmatrix} 243.979 & \\ 0 & .0164 \end{pmatrix}$	$\begin{pmatrix} 300.221 & \\ 0 & .0440 \end{pmatrix}$
$\hat{\boldsymbol{\Sigma}}_2$	$\begin{pmatrix} 119.15 & \\ .6376 & .0068 \end{pmatrix}$	$\begin{pmatrix} 243.979 & \\ 0 & .0164 \end{pmatrix}$	$\begin{pmatrix} 300.2213 & \\ 0 & .0440 \end{pmatrix}$
$\hat{\boldsymbol{\Sigma}}_3$	$\begin{pmatrix} 153.92 & \\ .6535 & .0062 \end{pmatrix}$	$\begin{pmatrix} 243.9788 & \\ 0 & .0164 \end{pmatrix}$	$\begin{pmatrix} 300.221 & \\ 0 & .0440 \end{pmatrix}$

Table 4.5: Average EM algorithm output values from the 142 simulations of fitting dat1 to 3-component mixtures of t-distributions, EM initialized via (c) k-means algorithm and (d) hierarchical clustering.

4.3.2 Simulated trivariate data from a 4-component mixture of *t*-distributions.

Let `dat2` denote a 3-dimensional simulated data set from a four component mixture of multivariate t-distributions with a common variance $\boldsymbol{\Sigma}$ across the components i.e $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \boldsymbol{\Sigma}$. The mixture distribution is assumed to have parameters:

$$\boldsymbol{\mu}_1 = (2.5, 2.8, 1.8)^T, \qquad \boldsymbol{\mu}_2 = (-2.5, -2.8, 3.8)^T$$
$$\boldsymbol{\mu}_3 = (0, 1.2, 0.5)^T, \qquad \boldsymbol{\mu}_4 = (3.0, -1.2, -1.5)^T$$
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & 0.5 & 0.1 \\ 0.5 & 1.0 & 0.3 \\ 0.1 & 0.3 & 1.0 \end{pmatrix}$$

$$\boldsymbol{\tau} = (0.25,\ 0.25,\ 0.25,\ 0,25)^T \qquad \text{and} \qquad \boldsymbol{\nu} = (4, 4, 4, 4)^T$$

Figure 4.4 shows the distribution of the variables in the simulated data set `dat2`. A Pairwise variables scatter plot of the simulated sample data is given Figure 4.6. Repeatedly fitting `dat2` to a 4-component mixture of multivariate t-distributions using 120 simulations for each initialization method, gives the average EM output presented in Table 4.6. The respective average values of the convergent log-likelihood, convergent error, AIC and BIC for the models fitted using `dat2` are shown in Table 4.6. For a comparison of the initialization methods based on average parameter estimates and the true parameter values for `dat2`, see Table 4.7 and Table 4.8.

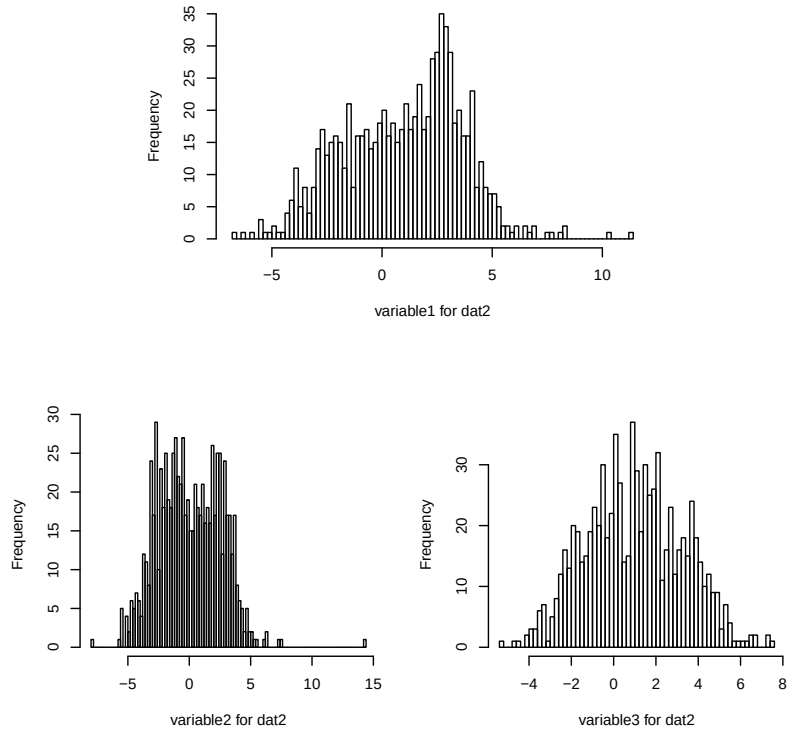

Figure 4.4: Distribution of variables in the simulated sample data dat2

	k-means	Random start	Hierarchical	Burn-in
`conv.error`	0	1	1	1
`BIC`	10079.35	9811.076	9811.075	9811.517
`AIC`	9991.715	9708.014	9708.014	9708.455
$l(\hat{\mathbf{\Psi}})$	-4978.857	-4832.007	-4832.007	-4832.228

Table 4.6: Average EM output values from the 120 simulations of fitting dat2 to mixtures of 4 t-distributions via EM algorithm initialized using (a) k-means algorithm, (b) random start method, (c) hierarchical clustering and (d) burn-in scheme.

A comparison of the convergent log-likelihoods is presented in Figure 4.5. Purple line represents the average convergent log-likelihood value from the EM algorithm initialized using hierarchical clustering which was the dominant method for dat2. The performance of the burn-in scheme and random start were almost as good as that of the hierarchical clustering methods while the k-means did not perform as good for the simulated data set `dat2`. The burn-in scheme compares favorably with the dominant hierarchical clustering method. The contours of fitting `dat2` to a mixture model featuring 4 t-distributions is shown in Figure 4.7.

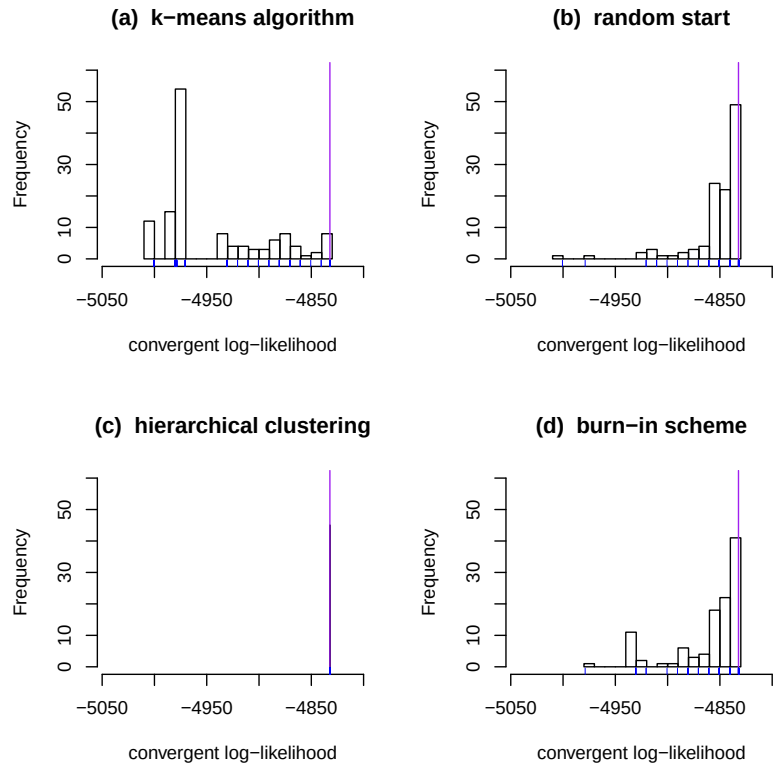

Figure 4.5: Distribution of Convergent log-likelihoods for 120 simulations of fitting dat2 to a 4-component mixture of t-distributions via the EM algorithm initialized using (a) k-means (b) random starts (c) hierarchical clustering and (d) burn-in.

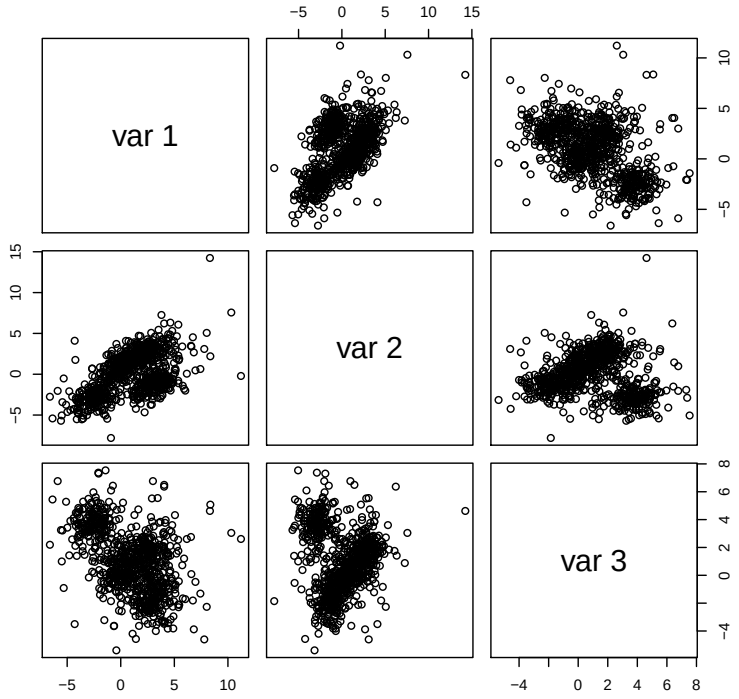

Figure 4.6: A variable pairwise scatter plot for the simulated data set 2

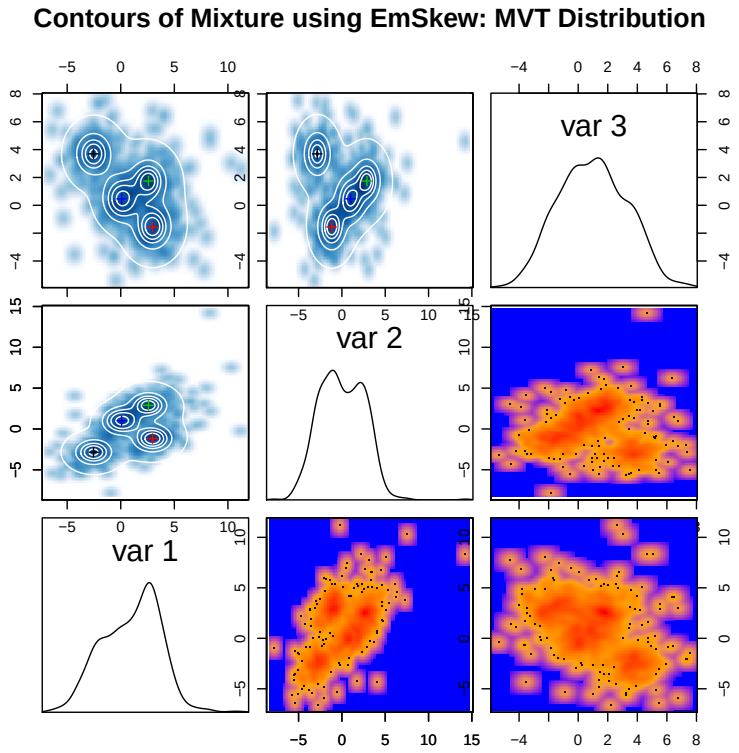

Figure 4.7: Contours for the simulated data set dat2 fitted via EM algorithm to a mixture of four t-distributions, with a common diagonal variance and using burn-in scheme as the initialization method.

Par.	True Par. Value	Par. Estimate When EM is Initialized via Random Start	Burn-in Scheme
$\hat{\boldsymbol{\tau}}$	$(.25, .25, .25, .25)^T$	$(.23, .27, .27, .23)^T$	$(.23, .27, .27, .23)^T$
$\hat{\boldsymbol{\nu}}$	$(4.0, 4.0, 4.0, 4.0)^T$	$(4.5, 4.3, 2.7, 53.0)^T$	$(4.5, 4.4, 2.7, 16.8)^T$
$\hat{\boldsymbol{\mu}}_1$	$(2.5, 2.8, 1.8)^T$	$(2.58, 2.89, 1.74)^T$	$(2.59, 2.89, 1.75)^T$
$\hat{\boldsymbol{\mu}}_2$	$(-2.5, -2.8, 3.8)^T$	$(-2.54, -2.85, 3.69)^T$	$(-2.54, -2.85, 3.70)^T$
$\hat{\boldsymbol{\mu}}_3$	$(0, 1.2, .5)^T$	$(.08, 1.01, .47)^T$	$(0.09, 1.01, 0.46)^T$
$\hat{\boldsymbol{\mu}}_4$	$(3.0, -1.2, -1.5)^T$	$(2.97, -1.18, -1.56)^T$	$(2.98, -1.18, -1.56)^T$
$\hat{\boldsymbol{\Sigma}}$	$\begin{pmatrix} 1.0 & & \\ .5 & 1.0 & \\ .1 & .3 & 1.0 \end{pmatrix}$	$\begin{pmatrix} 1.10 & & \\ 0 & .91 & \\ 0 & 0 & .95 \end{pmatrix}$	$\begin{pmatrix} 1.09 & & \\ 0 & .90 & \\ 0 & 0 & .94 \end{pmatrix}$

Table 4.7: Average parameter estimate values from the 120 simulations of fitting dat2 to mixtures of 4 multivariate t-distributions via the EM algorithm initialized using (a) random start method and (b) burn-in scheme.

Par.	True Par. Value	Par. Estimate When EM is Initialized via k-means Algorithm	Hierarchical clustering
$\hat{\boldsymbol{\tau}}$	$(.25, .25, .25, .25)^T$	$(.23, .27, .27, .23)^T$	$(.23, .27, .27, .23)^T$
$\hat{\boldsymbol{\nu}}$	$(4.0, 4.0, 4.0, 4.0)^T$	$(4.5, 4.3, 2.7, 50.3)^T$	$(4.5, 4.3, 2.7, 53.1)^T$
$\hat{\boldsymbol{\mu}}_1$	$(2.5, 2.8, 1.8)^T$	$(2.58, 2.89, 1.74)^T$	$(2.58, 2.89, 1.74)^T$
$\hat{\boldsymbol{\mu}}_2$	$(-2.5, -2.8, 3.8)^T$	$(-2.54, -2.85, 3.69)^T$	$(-2.54, -2.85, 3.69)^T$
$\hat{\boldsymbol{\mu}}_3$	$(0, 1.2, .5)^T$	$(0.081, 1.01, 0.465)^T$	$(0.081, 1.01, 0.47)^T$
$\hat{\boldsymbol{\mu}}_4$	$(2.97, -1.18, -1.56)^T$	$(3.0, -1.2, -1.5)^T$	$(2.97, -1.18, -1.56)^T$
$\hat{\boldsymbol{\Sigma}}$	$\begin{pmatrix} 1.0 & & \\ .5 & 1.0 & \\ .1 & .3 & 1.0 \end{pmatrix}$	$\begin{pmatrix} 1.10 & & \\ 0 & .91 & \\ 0 & 0 & .96 \end{pmatrix}$	$\begin{pmatrix} 1.10 & & \\ 0 & .91 & \\ 0 & 0 & .96 \end{pmatrix}$

Table 4.8: Average parameter estimate values from the 120 simulations of fitting dat2 to mixtures of 4 multivariate t-distributions via the EM algorithm initialized using (c) k-means algorithm and (d) hierarchical clustering.

4.3.3 Simulated bivariate data from a 3-component mixture of Gaussian distributions.

Let the simulated data in this subsection be referred to as `dat3`. Then data set `dat3` is a sample from a mixture of 3 Gaussian distributions with:

$$\boldsymbol{\mu}_1 = (2.5,\ 1.5)^T, \qquad \boldsymbol{\mu}_2 = (-2.5,\ -1.5)^T, \qquad \boldsymbol{\mu}_3 = (0,\ 5.2)^T$$

a common variance-covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3$ given by

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & \\ .5 & 1 \end{pmatrix}$$

and component membership parameter vector given by $\boldsymbol{\tau} = (0.5,\ 0.3,\ 0.2)^T$

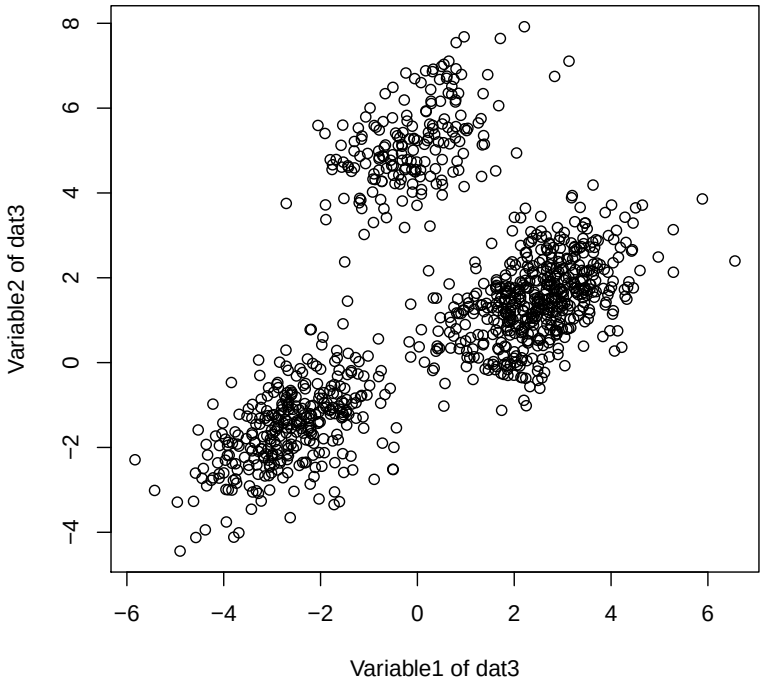

Figure 4.8: Scatter plot for 1000 data points in dat3

The sample data, `dat3` of size 1000 is shown in the scatter plot of Figure 4.8. Fitting `dat3` to a 3-component mixture of multivariate t-distributions gives the average values in Table 4.9. From the summary results in Table 4.9 based on 140 simulations, we see that the burn-in scheme gave the best average results for the simulated data set `dat3` since it gave the smallest average values of AIC and BIC. In general however, the difference in performance did not vary much across the different initialization methods. For example, the best model resulted from the EM algorithm initialized via the burn-in scheme and gave the average convergent log-likelihood value of $l(\hat{\boldsymbol{\Psi}}) = -3792.330$. The least preferred model came from the EM initialized via k-means and gave the average convergent log-likelihood value of

$l(\hat{\boldsymbol{\Psi}}) = -3792.685$. The distributions of the convergent log-likelihoods for the four initialization methods is presented in Figure 4.9. The purple line represents the average convergent log-likelihood value from the EM algorithm initialized using the hierarchical clustering algorithm.

	k-means	Random start	Hierarchical	Burn-in
`conv.error`	0	0	0	0
`BIC`	7675.170	7675.028	7674.904	7674.462
`AIC`	7611.361	7611.228	7611.108	7610.661
$l(\hat{\boldsymbol{\Psi}})$	-3792.685	-3792.614	-3792.551	-3792.330

Table 4.9: Average EM output values based on the 140 simulations of fitting dat3 to mixtures of 3 bivariate t-distributions via the EM algorithm initialized using (a) k-means (b) random start (c) hierarchical clustering and (d) burn-in scheme.

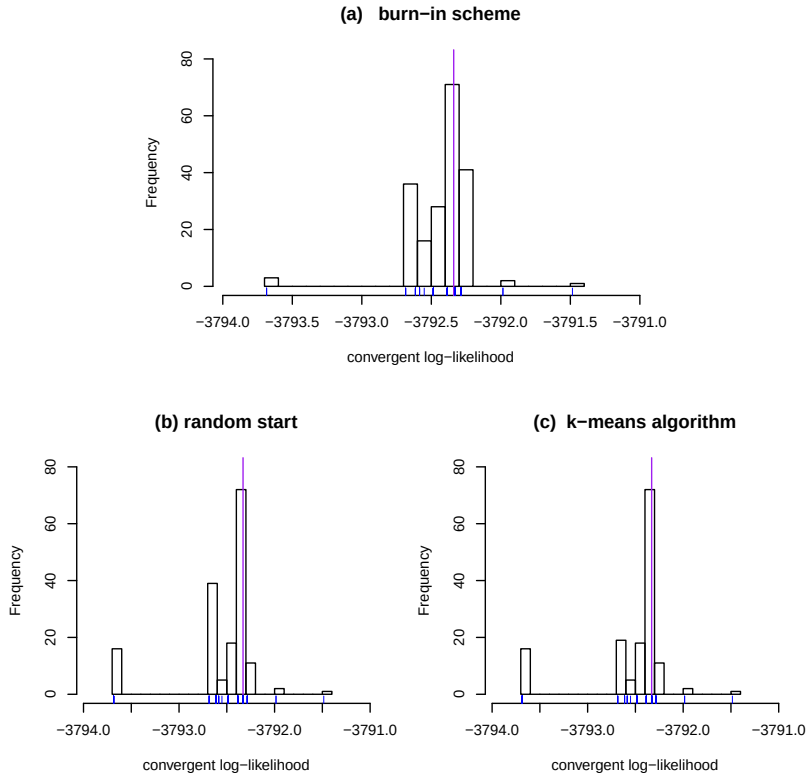

Figure 4.9: Distributions of Convergent log-likelihoods based on 140 simulations of fitting dat3 to a 2-component mixture of t-distributions via EM algorithm initialized using (a) burn-in scheme (b) random starts (c) k-means.

For further analysis, we compare the parameter estimates from the EM algorithm with the true parameter values. In Table 4.10, actual parameter (par.) values are compared with estimates from the EM algorithm initialized via random start and burn-in methods while in Table 4.11, par. values are

compared with estimates from the EM algorithm initialized via k-means and hierarchical clustering.

Par.	True Par. Value	Par. Estimate When EM is Initialized via Random Start	Burn-in Scheme
$\hat{\boldsymbol{\tau}}$	$(0.5, 0.3, 0.2)^T$	$(.50, .32, .18)^T$	$(0.495, 0.32, 0.185)^T$
$\hat{\boldsymbol{\nu}}$	unknown	$(10.2, 13.49, 10.2)^T$	$(12.07, 12.82, 12.39)^T$
$\hat{\boldsymbol{\mu}}_1$	$(2.5, 1.5)^T$	$(2.49, 1.47)^T$	$(2.49, 1.47)^T$
$\hat{\boldsymbol{\mu}}_2$	$(-2.5, -1.5)^T$	$(-2.59, -1.53)^T$	$(-2.58, -1.53)^T$
$\hat{\boldsymbol{\mu}}_3$	$(0, 5.2)^T$	$(-0.12, 5.14)^T$	$(-0.12, 5.15)^T$
$\hat{\boldsymbol{\Sigma}}$	$\begin{pmatrix} 1.0 & \\ .5 & 1.0 \end{pmatrix}$	$\begin{pmatrix} .8059 & \\ 0 & .7977 \end{pmatrix}$	$\begin{pmatrix} .8175 & \\ 0 & .809 \end{pmatrix}$

Table 4.10: Average EM output values based on 140 simulations of fitting dat3 to a 2-component mixture of multivariate t-distributions via the EM algorithm initialized via (a) random start and (b) burn-in methods.

Par.	True Par. Value	Par. Estimate When EM is Initialized via k-means Algorithm	Hierarchical clustering
$\hat{\boldsymbol{\tau}}$	$(0.5, 0.3, 0.2)^T$	$(.495, .185, .32)^T$	$(.50, .32, .18)^T$
$\hat{\boldsymbol{\nu}}$	unknown	$(9.72, 10.49, 10.47)^T$	$(9.99, 10.99, 11.31)^T$
$\hat{\boldsymbol{\mu}}_1$	$(2.5, 1.5)^T$	$(2.49, 1.47)^T$	$(2.49, 1.47)^T$
$\hat{\boldsymbol{\mu}}_2$	$(-2.5, -1.5)^T$	$(-2.59, -1.52)^T$	$(-2.59, -1.52)^T$
$\hat{\boldsymbol{\mu}}_3$	$(0, 5.2)^T$	$(-0.12, 5.14)^T$	$(-0.12, 5.15)^T$
$\hat{\boldsymbol{\Sigma}}$	$\begin{pmatrix} 1.0 & \\ .5 & 1.0 \end{pmatrix}$	$\begin{pmatrix} .7934 & \\ 0 & 0.7864 \end{pmatrix}$	$\begin{pmatrix} .7987 & \\ 0 & .7916 \end{pmatrix}$

Table 4.11: Average EM output values based on 140 simulations of fitting dat3 to a 2-component mixture of multivariate t-distributions via the EM algorithm initialized using (c) k-means and (d) hierarchical clustering.

The contours resulting from fitting `dat3` to a 3-component mixture of bivariate t-distributions is presented in Figure 4.10. The initialization here employs the burn-in scheme as it was the dominant method throughout the 140 simulations of fitting `dat3` to mixtures of t-distributions. The contours from this model is compared to the contours plot in Figure 4.11 resulting from fitting `dat3` to a 3-component mixture of bivariate t-distributions with $\boldsymbol{\Psi}^{(0)}$ in the later model selected using the hierarchical clustering as it was the second most competitive method. Hence, the contours for the two models serve as a comparison ground in the performance of the burn-in scheme to that of the hierarchical clustering algorithm. From Figure 4.10 and Figure 4.11, we can see that the burn-in scheme performs just as good as the hierarchical clustering algorithm as both models are able to effectively identify the groups through in the clustering solution for `dat3`.

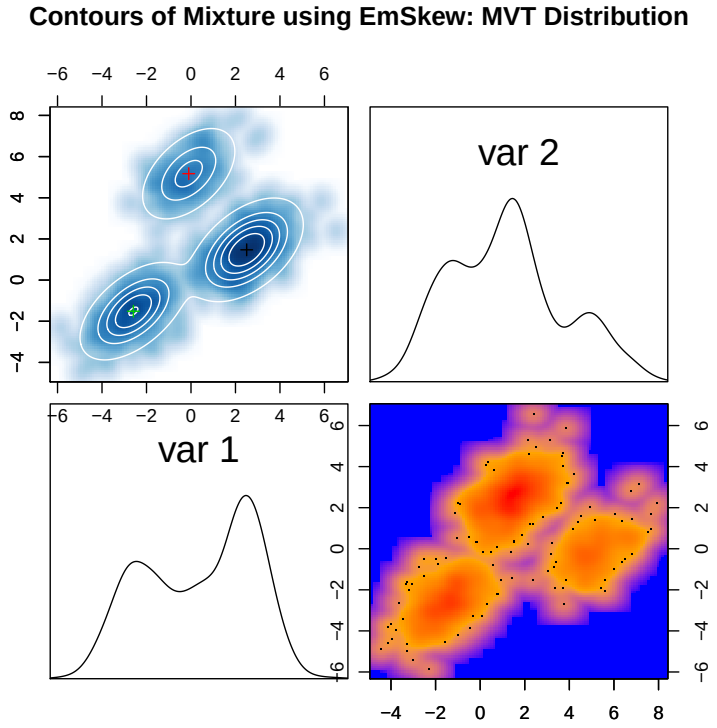

Figure 4.10: Contours for dat3 fitted via EM algorithm to a 3-component mixture of multivariate t-distributions with a general variance, using the burn-in concepts as the EM initialization method.

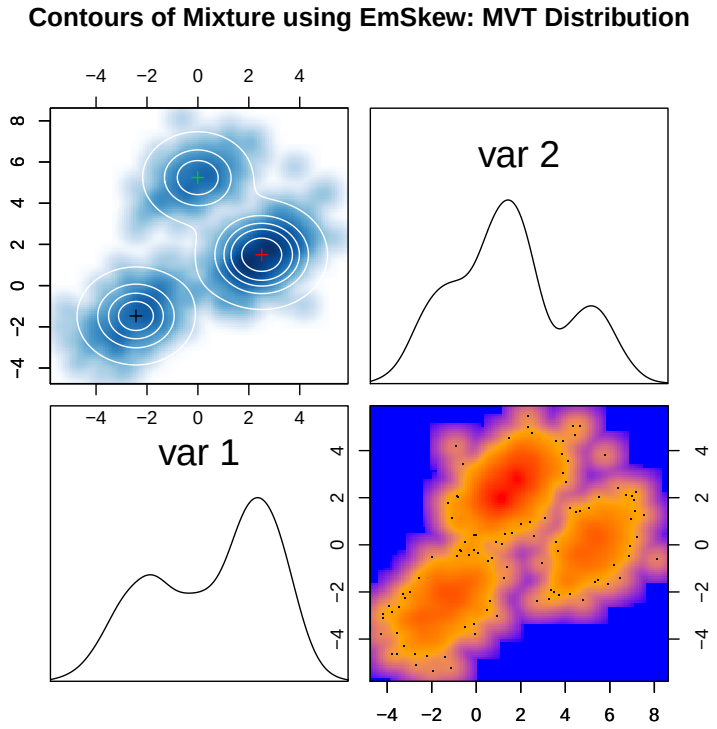

Figure 4.11: Contours for dat3 fitted via the EM algorithm to a mixture of three t-distributions with a common diagonal variance, EM starting point attained through hierarchical clustering.

4.3.4 Simulated trivariate data from a 4-component mixture of Gaussian distributions.

The simulated trivariate sample data, which we refer to as `dat4`, is obtained from a 4-component mixture distribution with four Gaussian components whose respective center parameters are:

$$\boldsymbol{\mu}_1 = (1.5,\ 1.8,\ 1.2)^T, \qquad \boldsymbol{\mu}_2 = (-1.5,\ -1.8,\ -1.2)^T$$
$$\boldsymbol{\mu}_3 = (0,\ -1.2,\ 1.2)^T, \qquad \boldsymbol{\mu}_4 = (0.5,\ 2.1,\ 1.2)^T,$$

a common covariance matrix across the four components given by:

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & & \\ 0.5 & 1.0 & \\ 0.1 & 0.2 & 1.0 \end{pmatrix}$$

and a component indicator vector given by:

$$\boldsymbol{\tau} = (0.25,\ 0.25,\ 0.25,\ 0,25)^T$$

The simulated data set (`dat4`) is shown in the variable pairwise scatter plots of Figure 4.13 and the distribution of the respective variables are shown in Figure 4.12. Repeatedly fitting this data set to a 4-component mixture of multivariate t-distributions via the EM algorithm initialized using (a) k-means algorithm (b) random starts (c) hierarchical clustering and (d) burn-in scheme, gives a summary of the averages in Table 4.12.

	k-means	Random start	Hierarchical	Burn-in
`conv.error`	0	0	0	0
`BIC`	8234.4	8203.271	8203.271	8205.874
`AIC`	8154.761	8100.21	8100.21	8102.813
$l(\hat{\boldsymbol{\Psi}})$	-4060.381	-4028.105	-4028.103	-4029.406

Table 4.12: A summary of the average EM output values for dat4 using the four initialization methods indicated. These are average values from 100 simulations of fitting dat4 to a 4-component mixture model featuring multivariate t-distributions via the EM algorithm.

For the simulated data set `dat4`, the optimal initial point $\boldsymbol{\Psi}^{(0)}$ was generated using the hierarchical clustering and the random start methods. This was one of the few instances when the random start method outperformed both the k-means algorithm and the burn-in scheme. Based on the convergent log-likelihood values, AIC values and BIC values, the most unpromising method when using the `dat4` data set was the k-means algorithm with an average convergent log-likelihood of $l(\hat{\boldsymbol{\Psi}}) = -4060.381$. The burn-in scheme performed fairly well for `dat4` giving an average convergent log-likelihood of $l(\hat{\boldsymbol{\Psi}}) = -4029.406$.

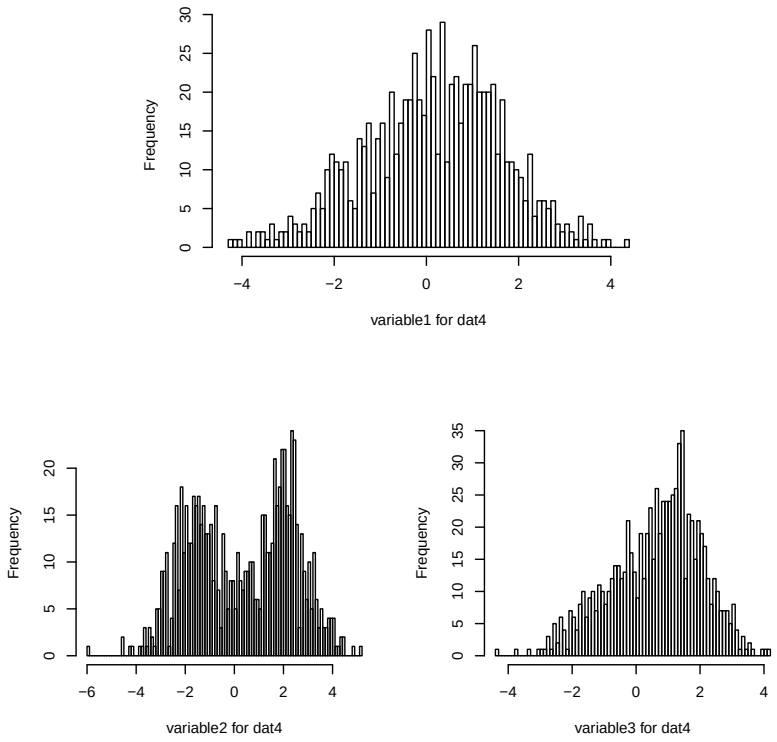

Figure 4.12: Histograms for the variables in the simulated data set (dat4).

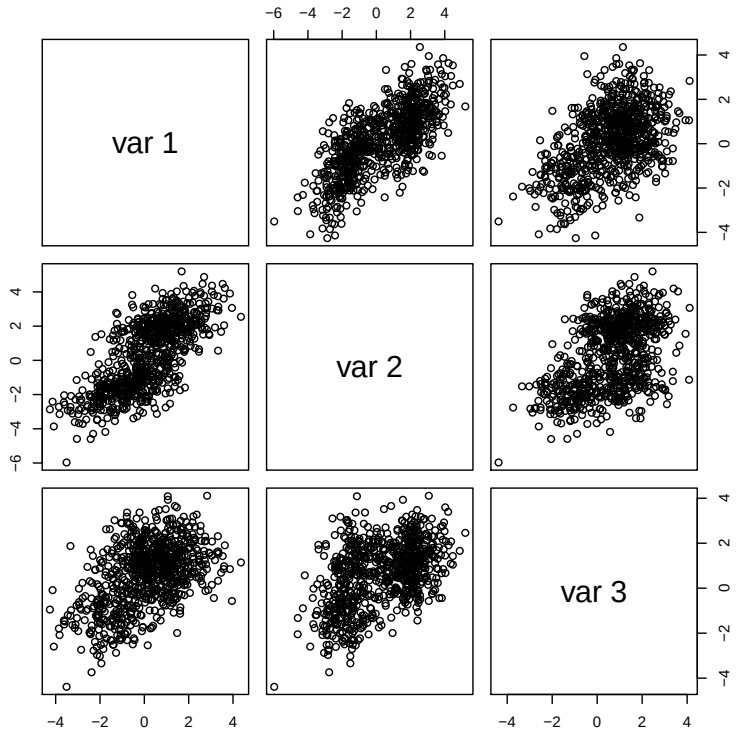

Figure 4.13: Variables pair plot for the variables in simulated data set(dat4).

We compare the parameter estimates from the EM algorithm with the true parameter values in Tables 4.13 and Table 4.14.

Par.	True Par. Value	Par. Estimate When EM is Initialized via Random Start	Par. Estimate When EM is Initialized via Burn-in Scheme
$\hat{\boldsymbol{\tau}}$	$(.25, .25, .25, .25)^T$	$(.23, .36, .24, .16)^T$	$(.23, .37, .24, .16)^T$
$\hat{\boldsymbol{\nu}}$	unknown	$(13, 200, 149, 200)^T$	$(13, 190, 106, 200)^T$
$\hat{\boldsymbol{\mu}}_1$	$(1.5, 1.8, 1.2)^T$	$(1.9, 2.8, 1.5)^T$	$(1.9, 2.8, 1.5)^T$
$\hat{\boldsymbol{\mu}}_2$	$(-1.5, -1.8, -1.2)^T$	$(-1.6, -2.0, -1.1)^T$	$(-1.6, -2.0, -1.1)^T$
$\hat{\boldsymbol{\mu}}_3$	$(0, -1.2, 1.2)^T$	$(-0.1, -1.1, 1.1)^T$	$(-0.1, -1.1, 1.1)^T$
$\hat{\boldsymbol{\mu}}_4$	$(0.5, 2.1, 1.2)^T$	$(0.7, 1.6, 1.1)^T$	$(0.7, 1.6, 1.0)^T$
$\hat{\boldsymbol{\Sigma}}$	$\begin{pmatrix} 1 & & \\ .5 & 1 & \\ .1 & .2 & 1 \end{pmatrix}$	$\begin{pmatrix} .79 & & \\ 0 & .72 & \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} .79 & & \\ 0 & .72 & \\ 0 & 0 & .99 \end{pmatrix}$

Table 4.13: Comparison of the actual model parameter (par.) values with the average parameter estimates based on 100 simulations of fitting dat4 to a mixture model featuring 4 multivariate t-distributions via the EM algorithm initialized via (a) random start and (b) burn-in scheme.

Par.	True Par. Value	Par. Estimate When EM is Initialized via k-means Algorithm	Par. Estimate When EM is Initialized via Hierarchical clustering
$\hat{\boldsymbol{\tau}}$	$(.25, .25, .25, .25)^T$	$(.23, .37, .24, .16)^T$	$(.23, .37, .24, .16)^T$
$\hat{\boldsymbol{\nu}}$	unknown	$(13, 200, 139, 200)^T$	$(13, 200, 139, 200)^T$
$\hat{\boldsymbol{\mu}}_1$	$(1.5, 1.8, 1.2)^T$	$(1.9, 2.8, 1.5)^T$	$(1.9, 2.8, 1.5)^T$
$\hat{\boldsymbol{\mu}}_2$	$(-1.5, -1.8, -1.2)^T$	$(-1.6, -2.1, -1.1)^T$	$(-1.6, -2.0, -1.1)^T$
$\hat{\boldsymbol{\mu}}_3$	$(0, -1.2, 1.2)^T$	$(-.1, -1.1, 1.1)^T$	$(-.092, -1.1, 1.1)^T$
$\hat{\boldsymbol{\mu}}_4$	$(0.5, 2.1, 1.2)^T$	$(0.7, 1.6, 1.0)^T$	$(0.7, 1.6, 1.0)^T$
$\hat{\boldsymbol{\Sigma}}$	$\begin{pmatrix} 1 & & \\ .5 & 1 & \\ .1 & .2 & 1 \end{pmatrix}$	$\begin{pmatrix} .79 & & \\ 0 & .72 & \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.79 & & \\ 0 & .72 & \\ 0 & 0 & 1 \end{pmatrix}$

Table 4.14: Comparison of the actual model parameter (par.) values with the average parameter estimates based on 100 simulations of fitting dat4 to a mixture model featuring 4 multivariate t-distributions via the EM algorithm initialized via (c) k-means algorithm and (d) hierarchical clustering

4.4. Computational Results from Real Data Sets

In this section, we further investigate the performance of the burn-in initialization function by fitting some real data sets to mixtures of t-distributions via the EM algorithm. Six real data sets are used in this study; the `iris` data, `ais` data, `banknote` data, `faithful` data, `DLBCL` and `Lympho` data. A summary of these illustrative real data sets is presented in Table 4.15.

Data	Dimension (p)	Groups (g)	Sample size (n)
`iris`	4	3	150
`ais`	13	2	202
`banknote`	7	2	200
`faithful`	2	3	299
`DLBCL`	3	4	8000
`Lympho`	4	1	33399

Table 4.15: Summary of the real illustrative data sets

4.4.1 Anderson's iris Data

The `iris` data of [36] is a sample of size $n = 150$ observations containing measurements in centimeters of the variables sepal length, sepal width, petal length and petal width, respectively, for 50 flowers from each of the 3 species of the iris flower; the setosa, versicolor and virginica. Figure 4.14 shows the scatter plot of the variables in the `iris` data and Figure 4.15 shows the frequency distribution of the respective variables.

When fitted to a mixture model via the EM algorithm, `iris` data is clustered into $g = 3$ groups where the groups represents the three species. We present a summary analysis of fitting this 4-dimensional data set to a finite mixture model featuring three components, each with a multivariate t-distribution, via the EM algorithm.

The results of fitting the `iris` data to a finite mixture model featuring multivariate t-distributions using a single k-means partition, a single random start, hierarchical clustering and the burn-in concepts are summarized in Table 4.16. Similarly, the results of fitting the `iris` data to mixtures of multivariate t-distributions using the best of the 10 k-means partitions and the best random start of the 10 random starts are given in Table 4.17. The results from the hierarchical clustering and the burn-in concepts are also included for comparison purposes. The presented results in Table 4.16 and Table 4.17 are the average EM out put values based on 120 simulations of fitting the `iris` data to mixtures of multivariate t-distributions via the EM algorithm. Hence, the values in Table 4.16 and Table 4.17 compare the average performances of the three initialization methods; the k-means algorithm, random start method, hierarchical clustering and burn-in scheme.

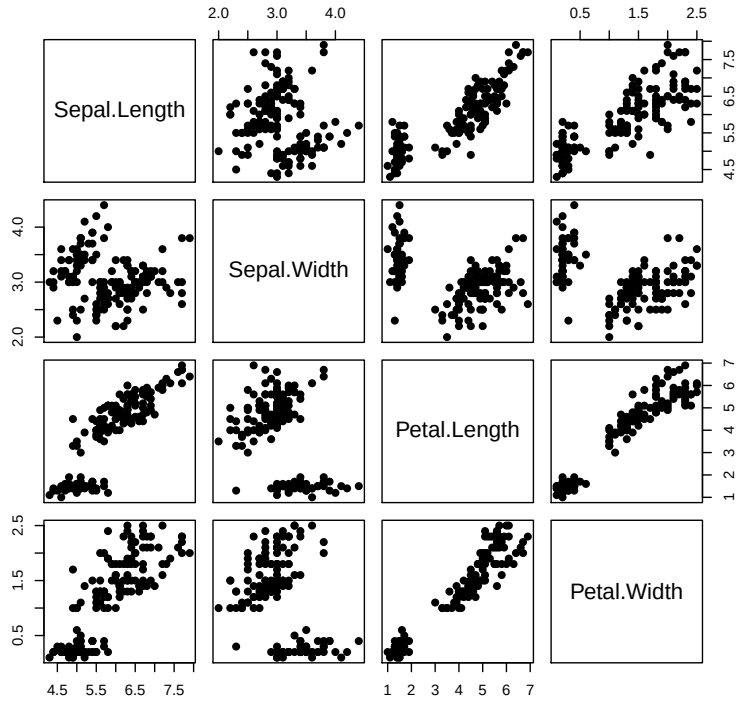

Figure 4.14: Pairwise variables plot for the iris data

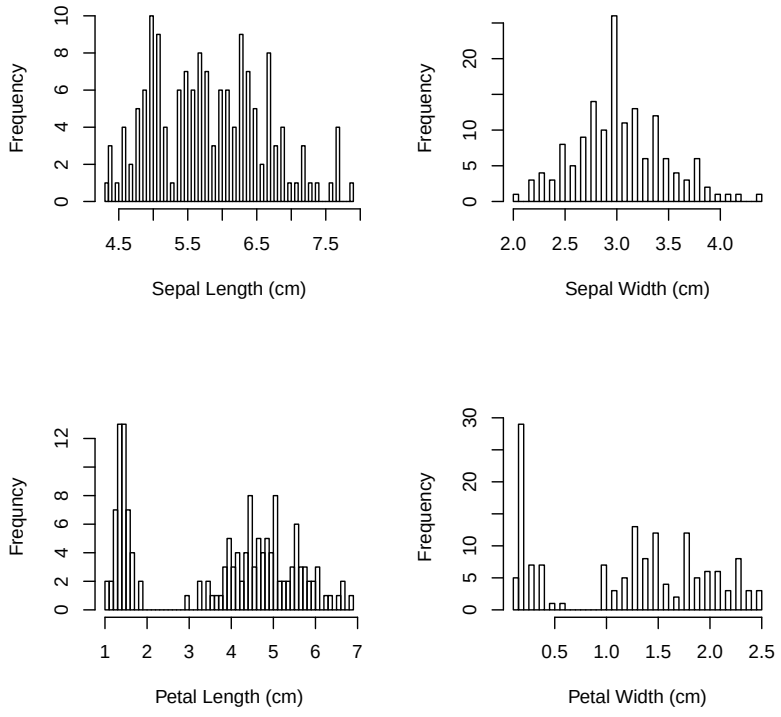

Figure 4.15: Frequency histograms of the variables in iris data

	$1k$-means	1Random start	Hierarchical	Burn-in
`conv.error`	0	0	0	0
`Error.rate`	0.047	0.047	0.047	0.047
`BIC`	793.3459	793.3457	793.3459	793.3459
`AIC`	730.1225	730.1223	730.1225	730.1225
$l(\boldsymbol{\Psi})$	-344.0613	-344.0612	-344.0613	-344.0613

Table 4.16: Average EM output values based on 120 simulations against initialization methods when the iris data is fit to a 3-component mixture of multivariate t-distributions via the EM algorithm.

	$10k$-means	10Random start	Hierarchical	Burn-in
`conv.error`	1	0	0	0
`Error.rate`	0.040	0.040	0.047	0.047
`BIC`	793.3456	793.3456	793.3459	793.3459
`AIC`	730.1223	730.1223	730.1225	730.1225
$l(\boldsymbol{\Psi})$	-344.0611	-344.0611	-344.0613	-344.0613

Table 4.17: Average EM output values based on 120 simulations against initialization methods when the iris data is fit to a 3-component mixture of multivariate t-distributions via the EM algorithm.

The optimal number of groups in the `iris` data set is $g = 3$. Based on the 120 simulations of fitting the `iris` data to a 3-component mixture of multivariate t-distributions via the EM algorithm, the largest attained average convergent log-likelihood value in our repeated experiments was $l(\hat{\boldsymbol{\Psi}}) = -344.0611$. This resulted from initializing the model parameters via repeated $10k$-means algorithm and repeated random start method. With the burn-in scheme approach, taking $b = 5$ resulted in a total of 32 initial candidate $\boldsymbol{Z}$ matrices for use in the burn-in scheme. The value of the average convergent log-likelihood under this scheme was $l(\hat{\boldsymbol{\Psi}}) = -344.0613$.

Table 4.16 and Table 4.17 present a summary of the average values from our repeated computations from the 120 simulations for each initialization method. From these tables, the k-means seems to have outperformed all other methods with a small margin in the average values of the convergent log-likelihoods. The main point emanating from Table 4.16 and Table 4.17 is that on average, the performance of the burn-in scheme compares favourably with that of the three dominant methods of EM initialization. In fact, with increasing number of simulations, the average performances of all the four methods seem to be identical.

Recall that when we are fitting continuous observed data sets to finite mixture distributions featuring multivariate t-distributions with $g-$components, we are actually allocating the observed data sets into g clusters based on

the component membership parameter $\boldsymbol{\tau}$ [3][4]. Therefore, if we know the true grouping of the data points into the respective groups (components) in a given data set (for example, see Figure 4.16), we can assess how good our model is through:

i) checking the number of miss-allocated data points in the fitted model by comparing with the known true allocation in the data.

ii) assessing the quality of the contours from the fitted model by comparing with the known true grouping of the data.

Recall that the `iris` data has size $n = 150$ data points comprising of $n_1 = 50$ `setosa`, $n_2 = 50$ `versicolor` and $n_3 = 50$ `virginica` species. This means that an accurate clustering of this data set into 3 groups requires that each group be allocated with exactly 50 data points. Any deviation from this clustering is considered to have an error. The error rate of any clustering result can be approximated by assessing the total number of miss-allocated data points. Table 4.18 shows a comparison of the final data clustering from the model fitted via EM algorithm using burn-in initialization, with the true class labels given by the variable `Species` in the `iris` data.

	Fitted Model Components			
	Comp.1	Comp.2	Comp.3	**Total**
`setosa`	50	0	0	50
`versicolor`	0	49	1	50
`virginica`	0	6	44	50
Total	50	55	45	150

Table 4.18: True class labels (Species) vs the final data clusters (components) from the model fitted via EM algorithm using burn-in initialization.

From Table 4.18, component 1 from our fitted model corresponds to the `setosa` group in the true grouping of the `iris` data, component 2 corresponds to the `versicolor` group in the true grouping of the `iris` data and component 3 corresponds to the `virginica` group in the true grouping of the `iris` data. Our fitted model manages to cluster all the `setosa` data points into one cluster. However, one `versicolor` data point is miss-allocated while six `virginica` data points are miss-allocated. A total of 7 observations out of 150 have been miss-allocated, giving an error rate of 4.7 percent (`Error.rate` = 0.047).

Similarly, Table 4.19 and Table 4.20 show data point allocation into three components when the k-means algorithm and the hierarchical clustering are used. Results from Table 4.19 are taken from the best of the 120 simulations of fitting `iris` to a mixture model via EM initialized using the k-means algorithm. Comparing values from Tables 4.18, 4.19 and 4.20, we conclude

that the four initialization methods show a similar performance, indicating that the proposed burn-in scheme will perform just as good as the main EM initialization methods such as the k-means.

	Fitted Model Components			
	Comp.1	Comp.2	Comp.3	**Total**
`setosa`	50	0	0	50
`versicolor`	0	49	1	50
`virginica`	0	5	45	50
Total	50	54	46	150

Table 4.19: True class labels (Species) vs the final data clustering (components) from the model fitted via EM algorithm using the k-means algorithm.

	Fitted Model Components			
	Comp.1	Comp.2	Comp.3	**Total**
`setosa`	50	0	0	50
`versicolor`	0	49	1	50
`virginica`	0	6	44	50
Total	50	55	45	150

Table 4.20: True class labels (species) vs the final data clustering (components) from the model fitted via EM algorithm using hierarchical clustering.

We can visualize and analyze the quality of a fitted model from the EM algorithm by assessing the quality of the contours from the fitted model. For example, Figure 4.17 shows the contours in the model resulting from fitting the `iris` data to a mixture of multivariate t-distributions with three components via the EM algorithm initialized using the burn-in scheme. The three component centers of the mixture model are indicated by the three coloured stars (red, green, blue) in Figure 4.17. Further, note that Figure 4.17 also shows the distribution of the variables in the fitted model as well as the heat maps for the distribution of the data points. The model shows that the variable sepal length has a multi-modal distribution, sepal width has a nearly symmetric unimodal distribution and the two variables petal length and petal width both have bimodal distributions.

By comparing Figure 4.17 with Figures 4.15 and 4.16, we can see that the fitted model closely depicts the true grouping and the respective distributions of the variables in the `iris` data. Figures 4.15 shows the frequency distribution of the four variables in the `iris` data while Figure 4.16 shows the true grouping in the data, as indicated by the categorical variable `Species` in the `iris` data.

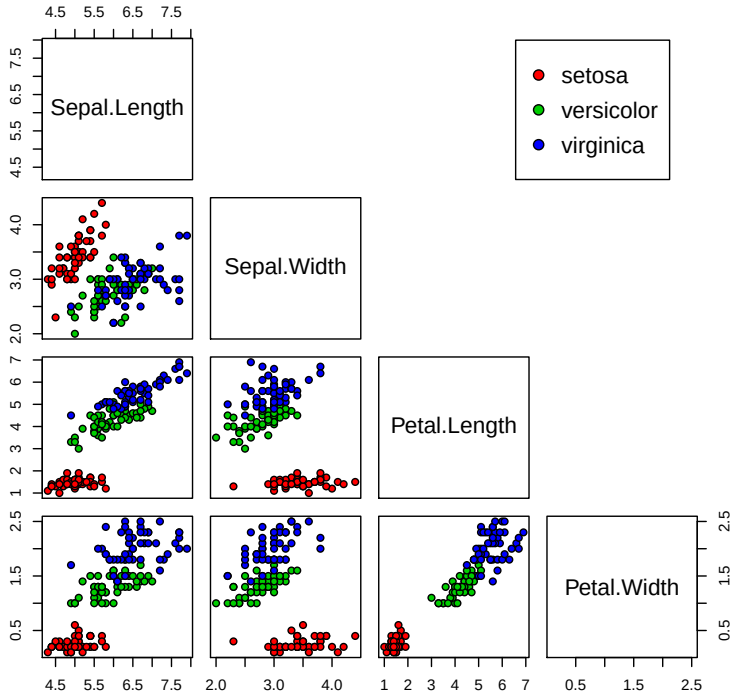

Figure 4.16: Pairs plot for the iris data showing the true grouping of data.

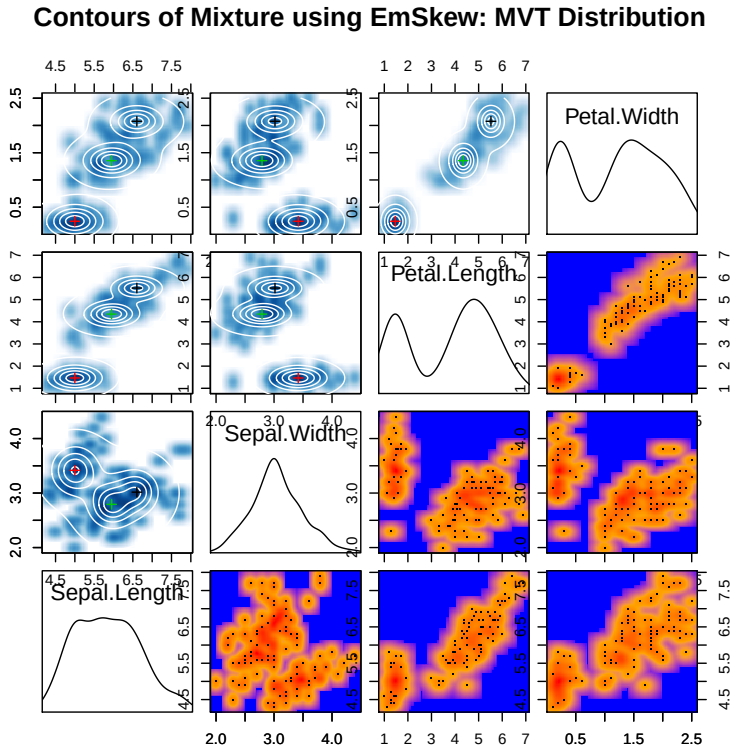

Figure 4.17: Contour plots for the iris data fitted to a 3-component mixture of t-distributions with a common diagonal variance and using the burn-in concepts as the initialization method.

Based on the 120 simulations of fitting the `iris` data to a 3-component mixture of multivariate t-distributions, we now present an analysis of the distribution of the convergent log-likelihood values from the EM algorithm presented in Figure 4.18. The red vertical line in the frequency distributions of Figure 4.18 represents the average optimum convergent log-likelihood value from EM algorithm initialized via repeated $10k$-means algorithm and repeated random start method, which was recorded at $l(\hat{\boldsymbol{\Psi}}) = -344.0611$. From the frequency histogram labeled (d), we see that the distribution of the values associated with the burn-in scheme does not show much variation and produces values close to the dominant mode (red vertical line) as is the case for both the k-means and the random start methods. This shows that the burn-in scheme can lead to a global mode in a high percentage of cases just like the other dominant methods. The hierarchical clustering method produced the same convergent log-likelihood value of $l(\hat{\boldsymbol{\Psi}}) = -344.0613$ in all trials unlike the other methods which yielded a variety yet very close convergent log-likelihood values. One point drawn from analyzing Figure 4.18 is that the average performance of the four initialization methods is almost identical.

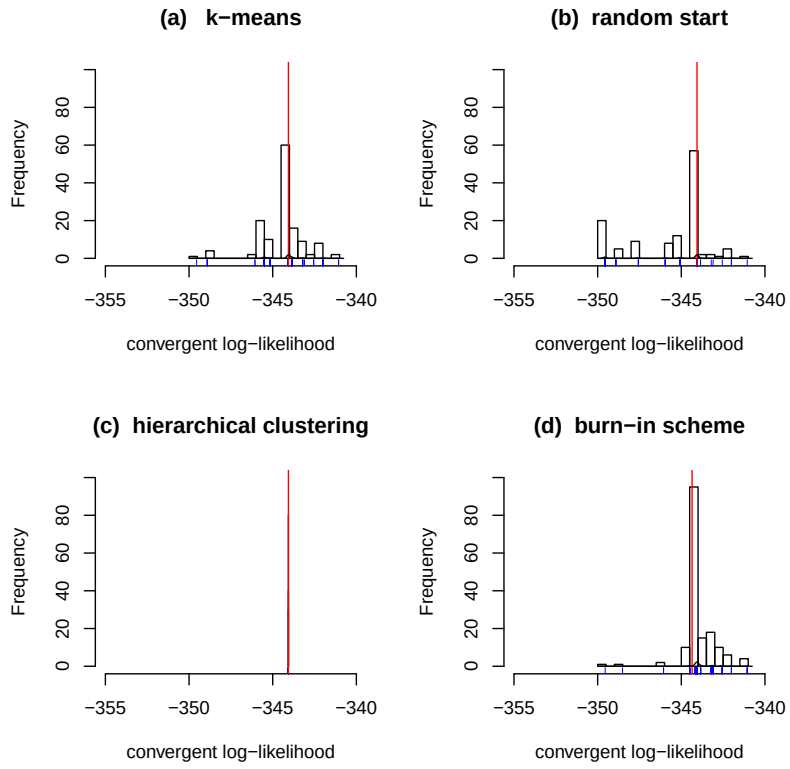

Figure 4.18: Distribution of the convergent log-likelihood values based on the 120 simulations of fitting the iris data set via the EM algorithm with EM initializations using (a) k-means algorithm (b) random starts method (c) hierarchical clustering and (d) burn-in scheme.

4.4.2 Old Faithful Geyser Data

This data set contains waiting times for the eruptions and the duration times of the eruptions for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA [41]. The two variables, eruption time (which measures the duration time of the eruption process), and the waiting (which measures the waiting time until the next eruption), were measured on each of the 299 individual eruptions, giving a data frame of size $n = 299$. The distribution is shown in Figure 4.19. The duration time for the eruptions can be classified as low, medium or high. Thus, fitting the `faithful` data to a finite mixture model results in a mixture model having three components.

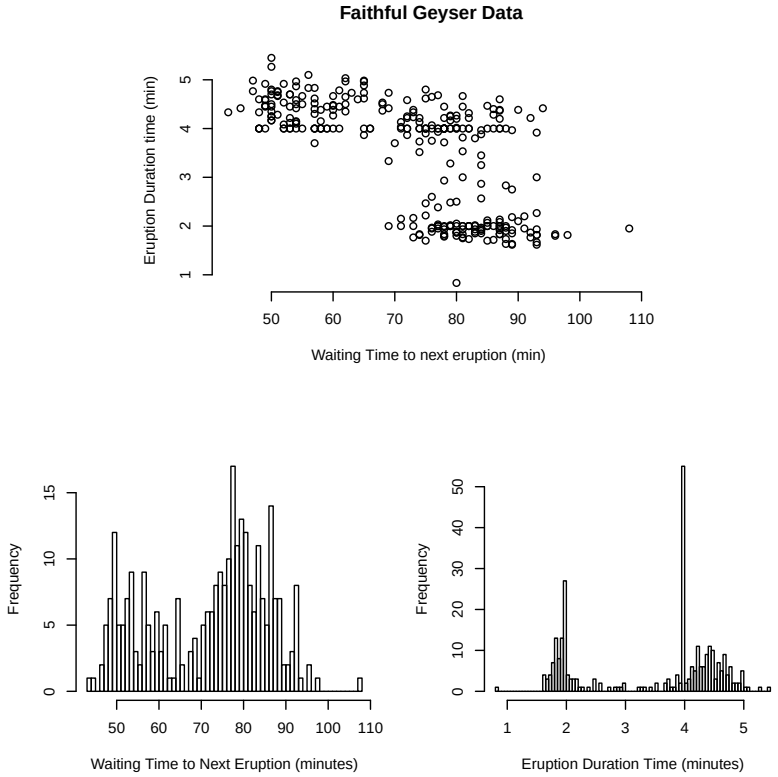

Figure 4.19: Scatter plot and histograms showing the distribution of the two variables in the faithful data.

A summary of average results from fitting the `faithful` data to a finite mixture model via the EM algorithm initialized using different methods are shown in Table 4.21. The average values from Table 4.21 are based on 100 simulations of fitting `faithful` data to finite mixtures of t-distributions with three components. The experimental results from this data set demonstrate that the burn-in scheme outperformed all the other initialization methods. The average values in the table show that the least performing initialization method was the hierarchical clustering method. The average performances of the repeated k-means, the random starts and the burn-in scheme did not show much differences as can be seen from their respective average convergent log-likelihood values presented Table 4.21.

	k-means	Random start	Hierarchical	Burn-in
`conv.error`	0	0	0	0
`BIC`	2808.025	2807.833	3188.896	2807.818
`AIC`	2759.919	2759.727	3140.790	2759.712
$l(\hat{\boldsymbol{\Psi}})$	-1366.959	-1366.864	-1557.395	-1366.856

Table 4.21: Summary of the average EM output values based on 100 simulations against initialization methods for the faithful geyser data fitted to a 3-component mixture of bivariate t-distributions.

Further, assessing the average convergent log-likelihood values, the highest values obtained under the action of the burn-in scheme gave a value of $l(\hat{\boldsymbol{\Psi}}) = -1366.856$ as can be seen from the summary values in Table 4.21. The distribution of the convergent log-likelihoods is given in Figure 4.20. The red vertical line represents the average optimum convergent log-likelihood value from EM algorithm initialized via the burn-in scheme. From Figure 4.20, we see that the burn-in gives a superior distribution of the convergent log-likelihoods as most values are clustered around the mode. This shows that the burn-in scheme will result in convergence to the global mode in a higher percentage of cases than any other initialization procedure.

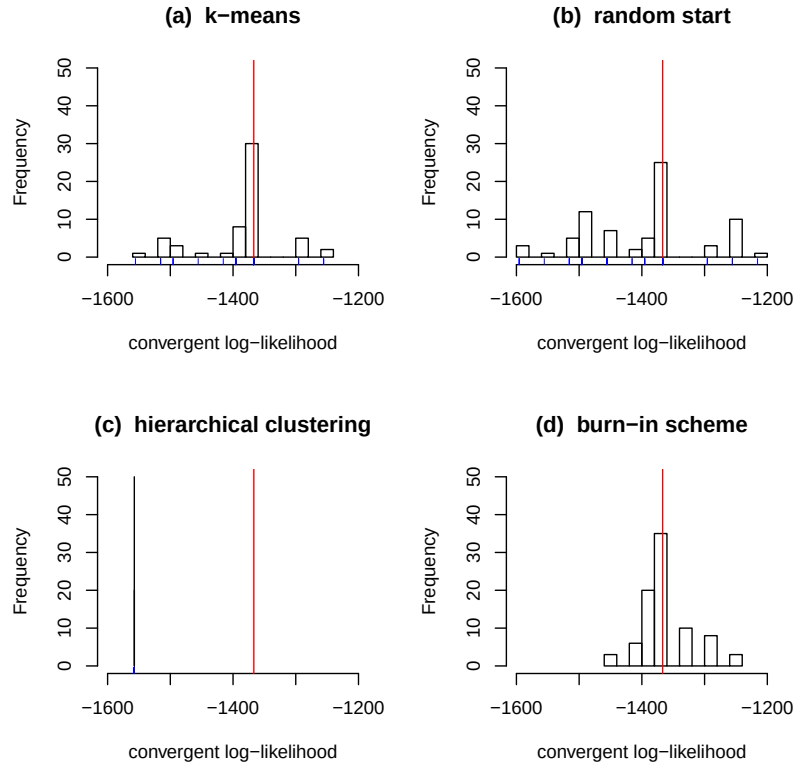

Figure 4.20: Distribution of convergent log-likelihood values for the faithful geyser data using (a) k-means algorithm (b) random starts (c) hierarchical clustering and (d) burn-in scheme.

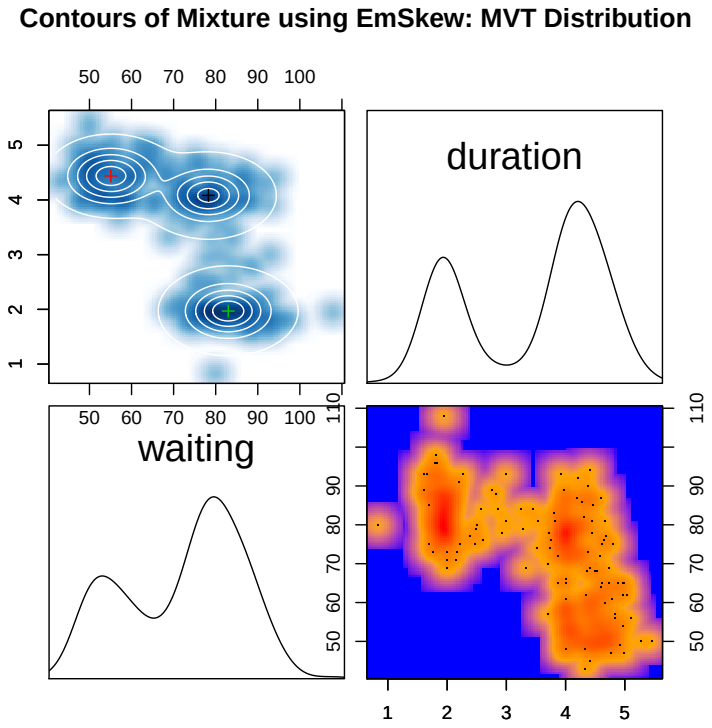

Figure 4.21: Contour plots for the faithful geyser data fitted to a 3-component mixture of t-distributions with a common diagonal variance and using the best of the ten k-means as the initialization method.

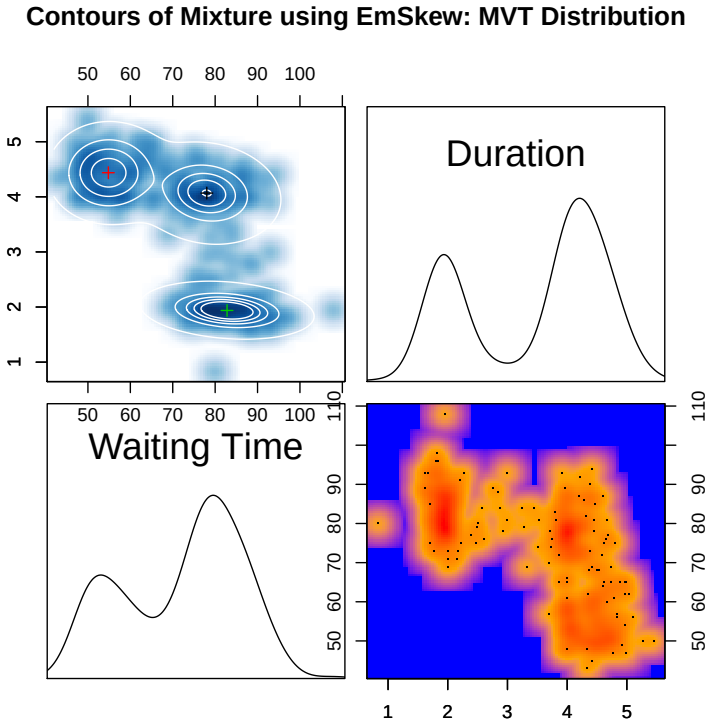

Figure 4.22: Contour plots for the faithful geyser data fitted to a 3-component mixture of t-distributions with a general variance and using the burn-in concepts as the initialization method.

4.4.3 Australian Institute of Sports (ais) Data

The `ais` data of [37], is a sample data of size $n = 202$ observation which contains biometric observations on 102 male and 100 female athletes collected at the Australian Institute of Sports, on the following variables: Sex (coded as 0 =male or 1 =female), Sport, Red Cell Count (RCC), White Cell Count (WCC), Hematocrit (Hc), Hemoglobin (Hg), Plasma Ferritin Concentration (Ferr), Body Mass Index (BMI), Sum of Skin Folds (SSF), Body Fat Percentage (Bfat), Lean Body Mass (LBM), Height (Ht in centimeters) and Weight (Wt in kilograms). A scatter plot of the pairs of the variables in the `ais` data set is shown in Figure 4.23. The discrete variables Sex and Sport are omitted resulting in a data matrix of dimension $p = 11$. The `ais` data can be fit to a mixture of $g = 2$ components, one component representing the male group and the other the female group.

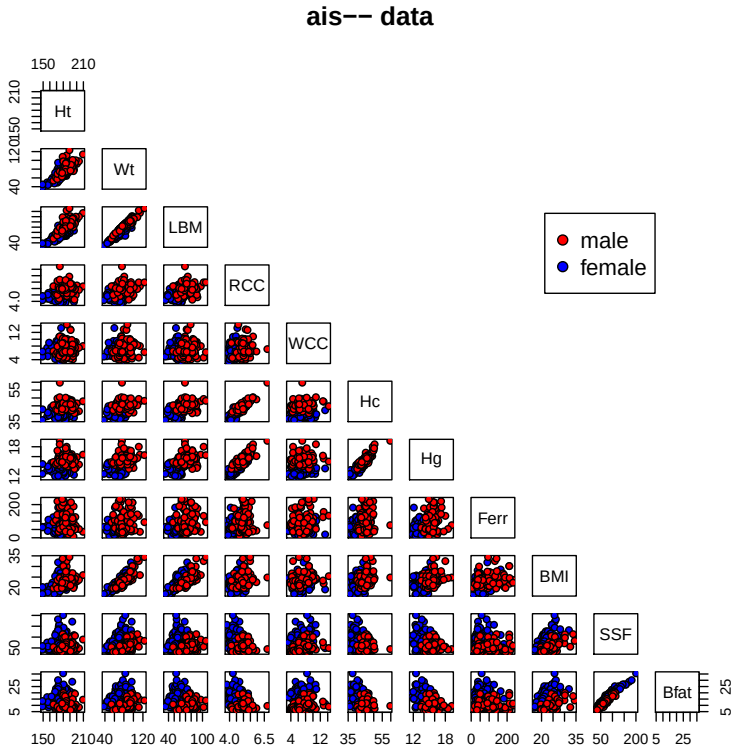

Figure 4.23: Pairwise variables plot with true grouping in the ais data set.

Table 4.22 shows the average EM algorithm results based on 200 simulations of fitting `ais` data to a 2-component mixture of multivariate t-distributions via the EM algorithm using various initialization methods. The average values from various initialization show little variation. In fact, as the number of simulations increases, the average performances of the four initialization methods is identical. These results are expected in cases where the likelihood function is not extremely multi-modal [4]. One important point drawn from this is that the burn-in scheme will perform just as good as any standard method of parameter initialization method, especially when the underlying likelihood function is not extremely multi-modal. The average convergent

log-likelihood in the fitted model for the `ais` data was $l(\hat{\boldsymbol{\Psi}}) = -6511.641$ for the burn-in initialization method, which did not show much variation from the results from other methods.

	k-means	Random start	Hierarchical	Burn-in
`conv.error`	0	0	0	0
`error rate`	0.034	0.034	0.034	0.034
`BIC`	13214.372	13214.377	13214.380	13214.361
`AIC`	13095.282	13095.289	13095.292	13095.277
$l(\boldsymbol{\Psi})$	-6511.647	-6511.656	-6511.661	-6511.641

Table 4.22: Average EM output values based on 200 simulations of fitting ais data to a 2-component mixture of t-distributions via EM algorithm.

Figure 4.24 shows the distribution of the convergent log-likelihoods associated with the various initialization methods. The red vertical line represents the average optimum convergent log-likelihood value from the EM algorithm initialized using the burn-in scheme with a value of $l(\hat{\boldsymbol{\Psi}}) = -6511.641$.

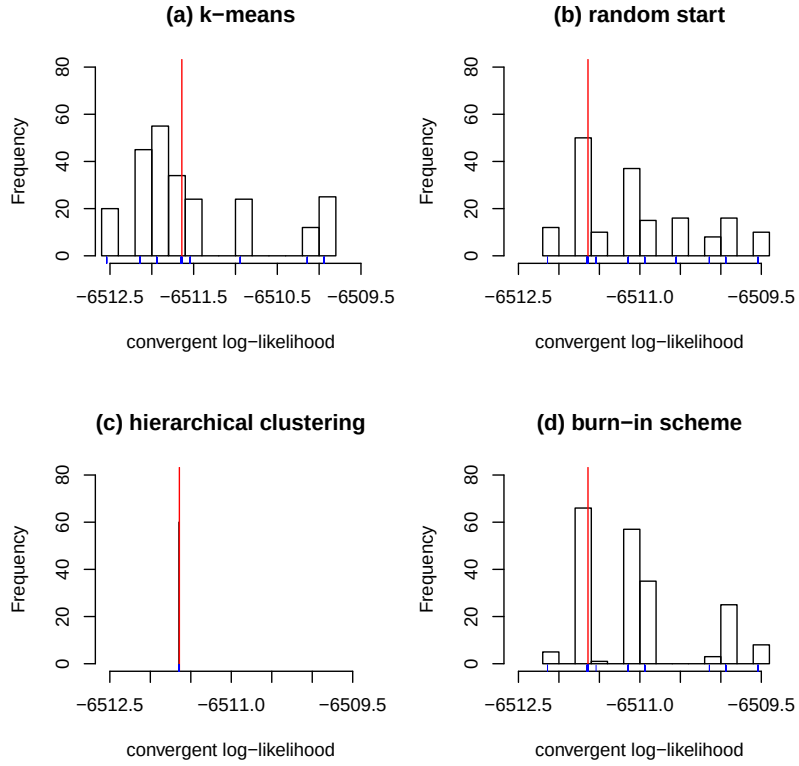

Figure 4.24: Distribution of convergent log-likelihoods for the ais data fitted to a 2-component mixture of t-distributions via the EM initialized using (a) k-means (b) random starts (c) hierarchical clustering and (d) burn-in.

4.4.4 Banknote Data

This data set contains six numerical variables measured (in millimeters) on 100 genuine and 100 counterfeit old-Swiss 1000-franc bank notes, and one qualitative variable (see [38]). The variables are: status (genuine or counterfeit), length (length of a note), left (width of left edge), right (width of right edge), bottom (bottom width), top (top width) and diagonal (Length of diagonal). This data can be fit to a $g = 2$ component mixture of multivariate t-distributions, each component representing status of the notes (genuine or counterfeit). Figure 4.25 and Figure 4.26 show distributions in the form of histograms for the variables in the `banknote` data and the variable pairwise scatter plots, respectively.

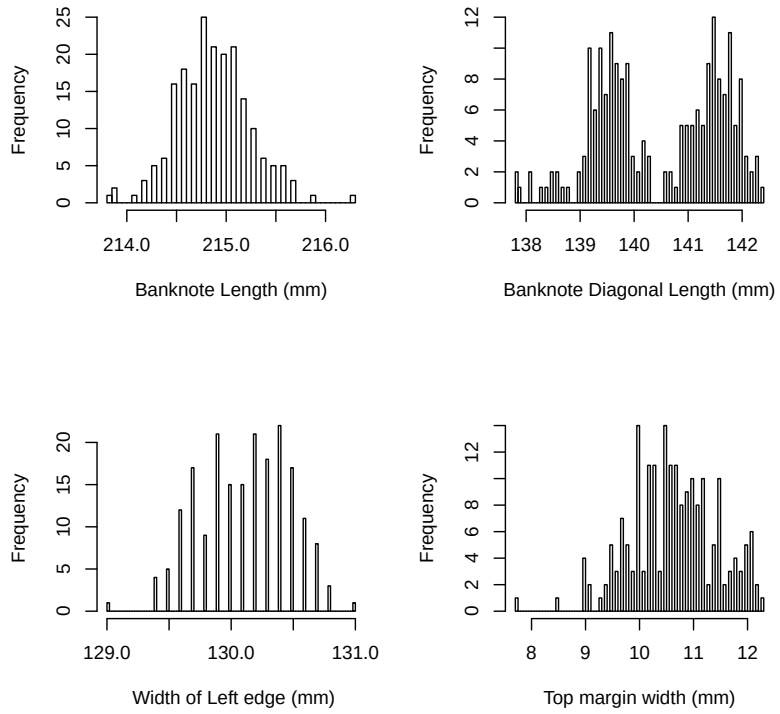

Figure 4.25: Distribution of variables in the Banknote data

We present a summary analysis of the results from the `banknote` data set. We fit a mixture of two components, one component representing the original notes and the other representing the counterfeit notes, available in the `banknote` data set. A comparison of the average EM output values for the different initialization methods based on 110 simulations is presented in Table 4.23 and Table 4.24.

Repeated simulations using the `banknote` data showed the burn-in scheme to be the dominant method for obtaining the starting points for the EM algorithm. Under the action of the burn-in function, the average value of the convergent log-likelihood based on 110 simulations of fitting the `banknote` data to a mixture of 2-component t-distributions was determined to be

$l(\hat{\boldsymbol{\Psi}}) = -906.0043$. The least performing method was the k-means algorithm giving an average convergent log-likelihood value of $l(\hat{\boldsymbol{\Psi}}) = -909.0177$.

Figure 4.27 shows the contour plots for the `banknote` data fitted to a 2-component mixture of t-distributions with a common diagonal variance using the burn-in concepts as the initialization method. These are compared with the true grouping in the data set based on the categorical variable `status` as shown in Figure 4.26. Comparing the two figures shows how accurate our model describes the distribution of these data sets.

Further, the distribution of the convergent log-likelihood values are shown in Figure 4.28. The red vertical line indicates the average modal value of $l(\hat{\boldsymbol{\Psi}}) = -906.0051$ from the EM algorithm initialized via the burn-in scheme. This value differs slightly from the average mean value of $l(\hat{\boldsymbol{\Psi}}) = -906.0043$ as can be seen from Tables 4.23 and 4.24 when the burn-in scheme is the initialization method employed. For the burn-in scheme, convergence to the mode occurred within the first 40 iterations of the EM algorithm showing a quick convergence (see Figure 4.29). Further, assessing Figure 4.28, we note that the burn-in scheme yielded a better distribution of the convergent log-likelihood values than any other approach. We conclude that with an increase in the number of simulations using `banknote` data, burn-in scheme proved to be the dominant method for obtaining $\boldsymbol{\Psi}^{(0)}$.

	1k-means	1Random start	Hierarchical	Burn-in
`conv.error`	0	0	0	0
`error.rate`	0.025	0.01	0.01	0.01
`BIC`	1923.383	1923.2734	1923.2738	1923.2731
`AIC`	1854.118	1854.0087	1854.0091	1854.0081
`conv.`$l(\boldsymbol{\Psi})$	-909.137	-906.0044	-906.0045	-906.0043

Table 4.23: Summary of average EM output values based on 110 simulations against initialization methods for the banknote data set.

	10k-means	10Random start	Hierarchical	Burn-in
`conv.error`	0	0	0	0
`error.rate`	0.02	0.01	0.01	0.01
`BIC`	1923.3000	1923.2824	1923.2738	1923.2731
`AIC`	1854.0554	1854.0177	1854.0091	1854.0081
$l(\boldsymbol{\Psi})$	-909.0177	-906.0089	-906.0045	-906.0043

Table 4.24: Summary of average output values from the EM algorithm based on 110 simulations against initialization methods for the banknote data set.

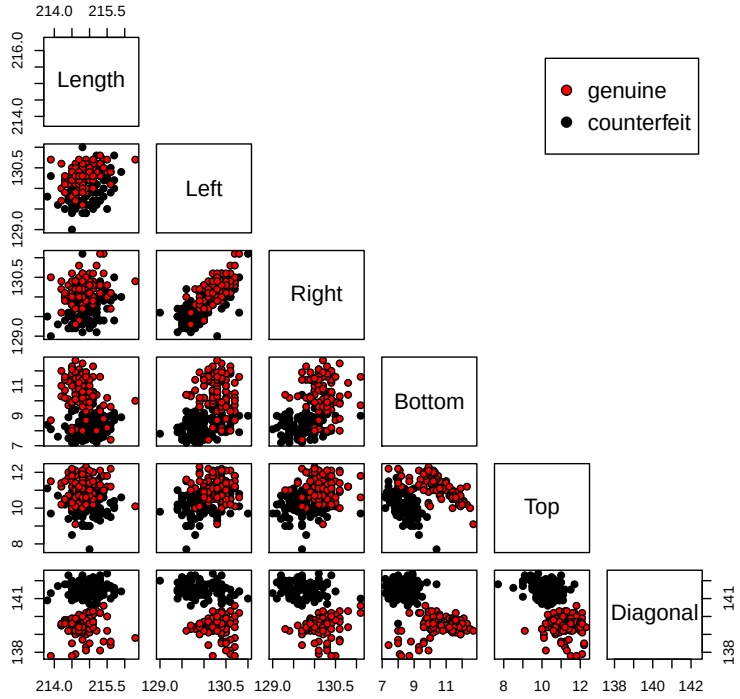

Figure 4.26: A variable pairwise scatter plot for the banknote data set showing the true grouping of the data sets.

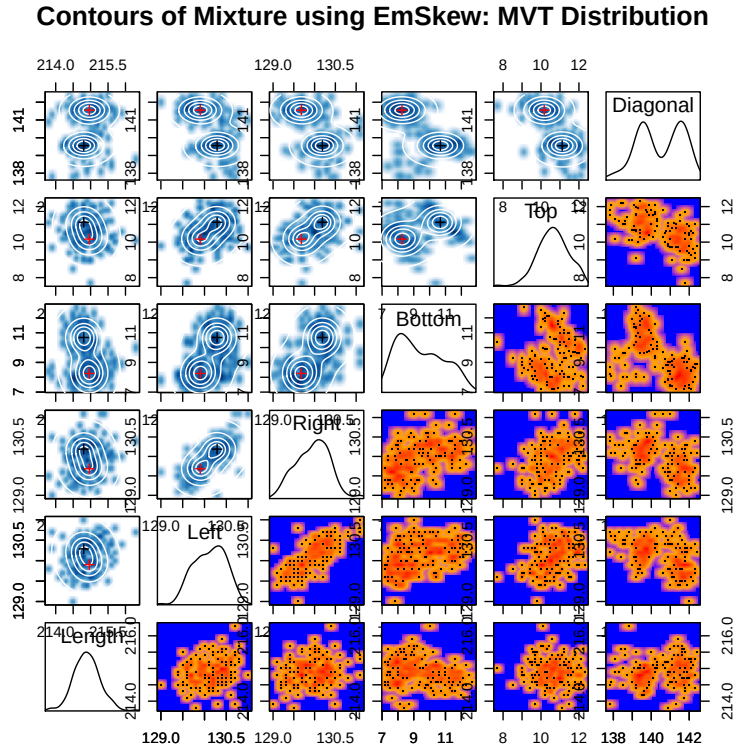

Figure 4.27: Contour plots for the banknote data fitted to a 2-component mixture of t-distributions with a common diagonal variance using the burn-in concepts as the initialization method.

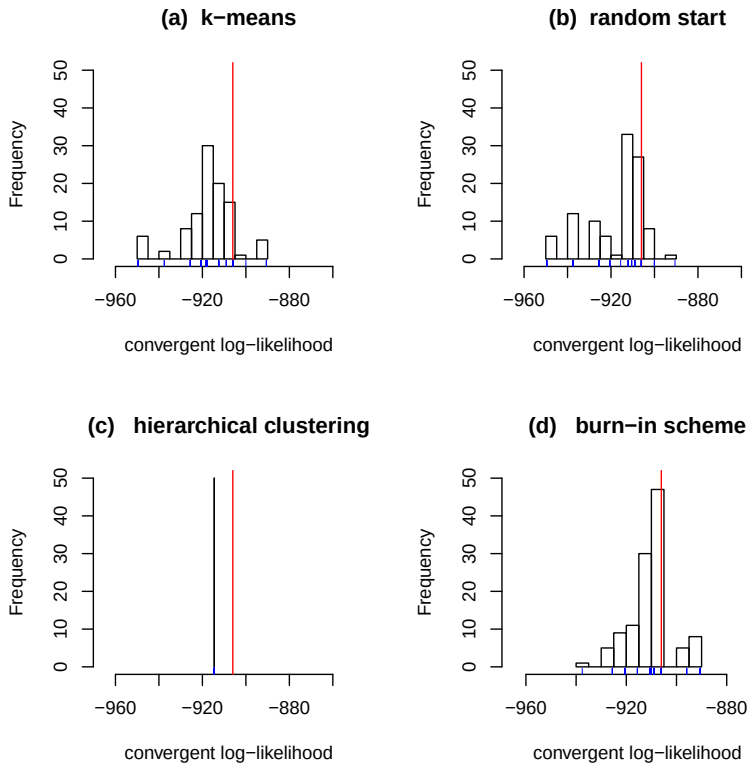

Figure 4.28: Distribution of Convergent log-likelihoods for the banknote fitted to a mixture of two t-distributions via the EM initialized using (a) k-means (b) random starts (c) hierarchical and (d) burn-in scheme.

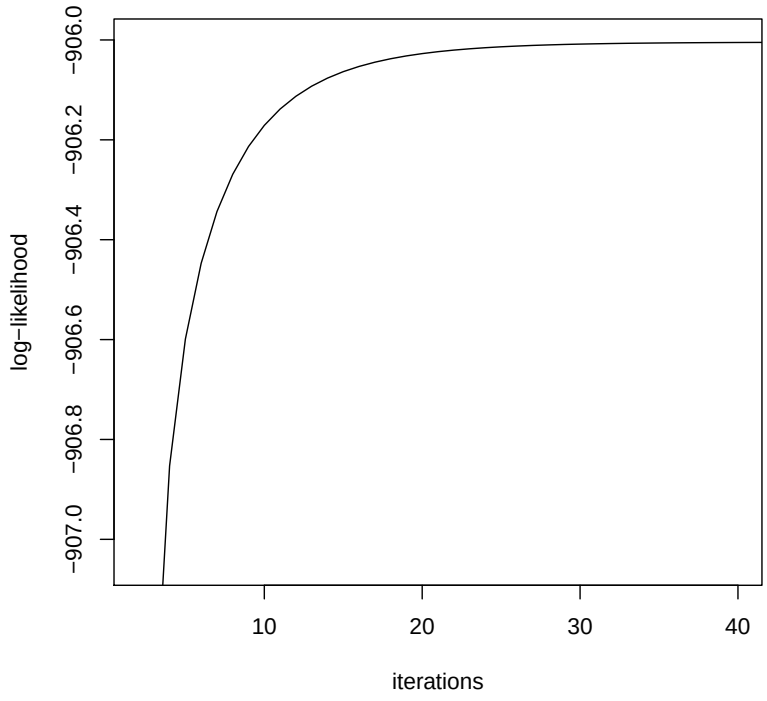

Figure 4.29: log-likelihood plot vs iterations for the banknote data fitted to a mixture of two t-distributions via EM initialized using burn-in scheme.

4.4.5 The Lymphoma Data Sets

One of the many important applications of finite mixture models is in the analysis of multidimensional cell populations through flow cytometric analysis [40][42]. Flow cytometry is a rapid technique for determining surface antigens on the cells teased from lymph nodes and other masses with suspected lymphoma [42]. It is a good technique in detecting and measuring many physical and chemical characteristics of a population of cells, making it helpful in detecting lymphoma affected cells in a given mass of tissue. The process involves identifying cells that are as similar as possible and then grouping them into clusters for component parameter estimation and subgroups identification [11].

In flow cytometric analysis, parallel measurements of fluorescent intensities are used to study the differential expression of different surfaces and intracellular proteins of a given blood sample [11]. The analysis typically involves identification of cell sub-populations (clusters) from the multidimensional data set, usually performed manually by visually separating regions (gates) of interests on a series of sequential bivariate projections of the data [40]. This process is known as gating. Due to the subjective and time-consuming nature of this approach and the difficulty in detecting higher-dimensional inter-marker relationships, many efforts have been made to develop computational methods to automate the gating process by automatically clustering the data from cells into a finite number of groups so that cells in a group are as similar as possible [42]. Mixture modelling using multivariate t-distribution finds its application in this process.

The two lymphoma data sets used in this study are the `Lympho` data set [42] and the `DLBCL` data set [39].

Lympho Data Set

The `Lympho` data set is a subset of the T-cell phosphorylation data set used in the grouping of cells (from body tissues suspected of having lymphoma) into several clusters [11][42]. The original data contain measurements of blood samples stained with four fluorophore-labeled antibodies against CD4, CD45RA, SLP76 and ZAP70. Measurements from each subject were taken before and after anti-CD3 stimulation. The data set used in this study is a subset of the pre-stimulation data from one subject. The two variables of interest are the SLP76 (marker 1) and ZAP70 (marker 2), which were measured on each of the 33399 observations (cells), giving a data frame of size $n = 33399$. A frequency histogram and smooth scatter plot are shown for this data set in Figure 4.30 and Figure 4.31, respectively. From Figure 4.30, we can see that both SLP76 (marker 1) and ZAP70 (marker 2) exhibit non-symmetric distributions. Each variable has a distribution that is skewed to the left about the mode.

In this study, we take $g = 2$ and $p = 2$ for the selected sample from the original `Lympho` data set and fit the data set to a mixture of $g = 2$ components of multivariate t-distribution.

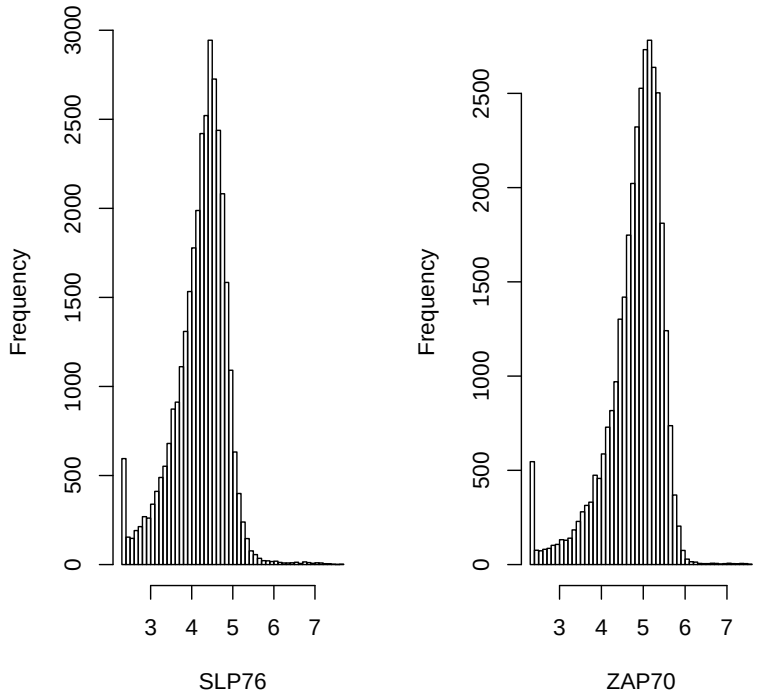

Figure 4.30: Distribution of the two variables in the Lympho data set.

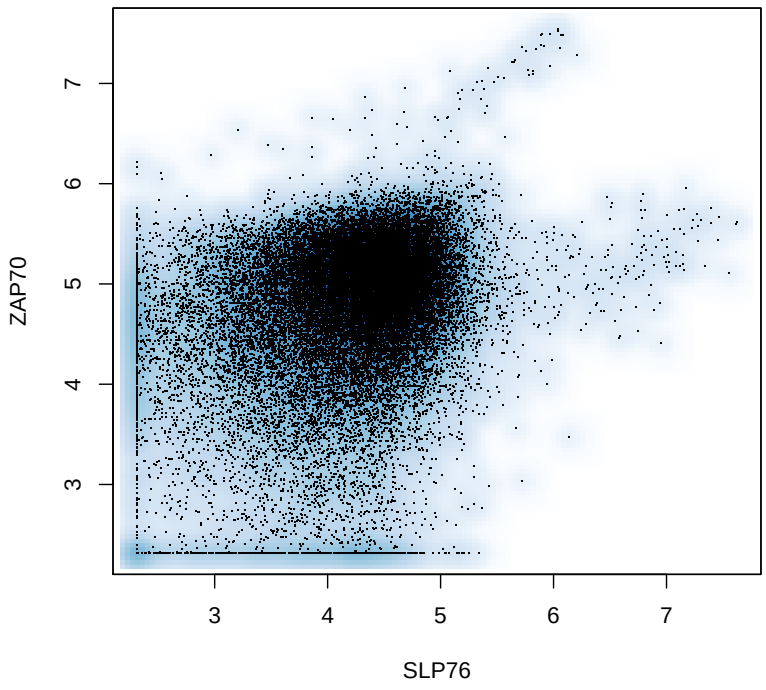

Figure 4.31: A smooth scatter plot for the Lympho data set.

From Table 4.15, recall that `Lympho` data set is of size $n = 33399$. However, computation of parameter estimates via the EM algorithm using such a huge sample tends to be painfully slow in **R** [44]. Thus, we select a random sub-sample, `LymphosubSample` of size $n = 1000$ for our computations.

We fitted the `LymphosubSample` data set to a 2-component mixture probability model of bivariate t-distributions via the EM algorithm using the optimal data partition from the burn-in scheme and the k-means algorithm as the initialization methods. Table 4.25 presents a summary of the average results from 100 simulations of fitting the `Lympho` data set to a 2-component mixture model of bivariate t-distributions via the EM algorithm.

	Using Burn-in scheme	Using 1k-means algorithm
`error`	0	0
`BIC`	3802.888	3802.885
`AIC`	3758.718	3758.715
$l(\boldsymbol{\Psi})$	-1870.359	-1870.358

Table 4.25: Average EM output values based on 100 simulations for the burn-in and k-means initialization methods when Lymho data set is fit to a 2-component mixture of bivariate t-distributions.

From Table 4.25, we see that the two models appear to be identical. This data set demonstrates again that the burn-in scheme will perform just as good as the dominant methods such as the k-means algorithm. A further analysis is made by comparing the parameter estimates. Table 4.26 presents average estimates from the two models. We can see that the estimates from the EM algorithm using burn-in scheme as the initialization method are just as good as those from the k-means initialized algorithm.

par.	Parameter(par.) Estimate When EM is Initialized via k-means Algorithm	Burn-in scheme
$\hat{\boldsymbol{\tau}}$	$(.81,\ .19,)^T$	$(.79,\ .21,)^T$
$\hat{\boldsymbol{\nu}}$	$(7, 2)^T$	$(10, 2)^T$
$\hat{\boldsymbol{\mu}}_1$	$(4.39, 4.96)^T$	$(4.39, 4.97)^T$
$\hat{\boldsymbol{\mu}}_2$	$(3.26, 4.35)^T$	$(3.32, 4.32)^T$
$\hat{\boldsymbol{\Sigma}}_{\mathbf{1}}$	$\text{diag}(0.15735, 0.22219)$	$\text{diag}(0.16228, 0.21723)$

Table 4.26: Average parameter estimates based on 100 simulations of the EM algorithm initialized via (a) k-means algorithm and (b) burn-in scheme.

The contours of fitting `Lympho` data to a two component mixture of bivariate t-distributions via the EM algorithm, is shown in Figure 4.32. The initial-

ization employed is the burn-in scheme as it was the best method of the four methods. However, we can clearly see from the contours in Figure 4.32 that the `Lympho` data set does not fit well to a two component model as the two groups are not effectively distinguished. One of the components is actually a subset of the other. This is not surprising because the distribution of the two variables in the `Lympho` data set show some skewness and the distributions are unimodal (see Figure 4.30). An ideal approach would be to consider a finite mixture model whose components are modelled using skewed distributions [11][12].

We consider an alternative model fitting the `Lympho` data set to a single component model featuring a bivariate t-distribution as shown in Figure 4.33. A comparison of Figure 4.32 and Figure 4.33 shows that a single component model is more appropriate for fitting the `Lympho` data set. We initialize the EM algorithm using both the burn-in scheme and the k-means algorithm so as to compare the resulting average EM out put values based on 50 simulations. These average results of fitting the `Lympho` data set to a single component model featuring a bivariate t-distribution are summarized in Table 4.27. Based on the values of the average convergent log-likelihood values from Table 4.27, we conclude that the burn-in scheme significantly outperformed the k-means algorithm in initializing the EM algorithm.

One important point from this analysis is that even in settings with data that is skewed (like `Lympho` data), the performance of the burn-in scheme rarely hurts. Hence, the burn-in scheme can be used as an optimal method of EM initialization with great success.

par.	$\hat{\boldsymbol{\tau}}$	$\hat{\boldsymbol{\nu}}$	$\hat{\boldsymbol{\mu}}$	$\hat{\boldsymbol{\Sigma}}$	$l(\boldsymbol{\Psi})$
(a) burn-in	1	5	$(4.78, 5.49)^T$	$\begin{pmatrix} .10155 & \\ .04759 & .05789 \end{pmatrix}$	-1800.639
(b) k-means	1	5	$(4.29, 4.88)^T$	$\begin{pmatrix} .23923 & \\ .06349 & .26930 \end{pmatrix}$	-1872.384

Table 4.27: Average parameter estimates based on 50 simulations, when the Lympho data set is fit to a single component model of a bivariate skewed t-distribution with EM initialized using (a) burn-in scheme (b) k-means.

In summary, the burn-in scheme has demonstrated to be an effective method of initializing the parameters in the EM algorithm when fitting the skewed `Lympho` data set to a single component model featuring a bivariate t-distribution. This is evidenced by the resulting contours in Figure 4.33, which are compared to the less promising contours of fitting `Lympho` data set to a two component model featuring bivariate t-distributions shown in Figure 4.32.

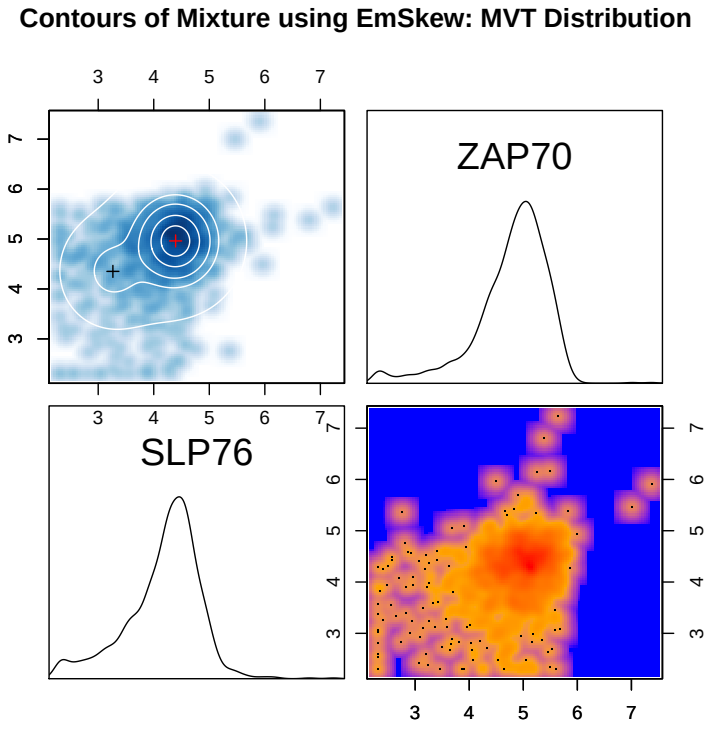

Figure 4.32: Contours for Lympho data set fitted to a 2-component mixture of bivariate t-distributions with a common diagonal variance via the EM algorithm, using the burn-in scheme as the initialization method.

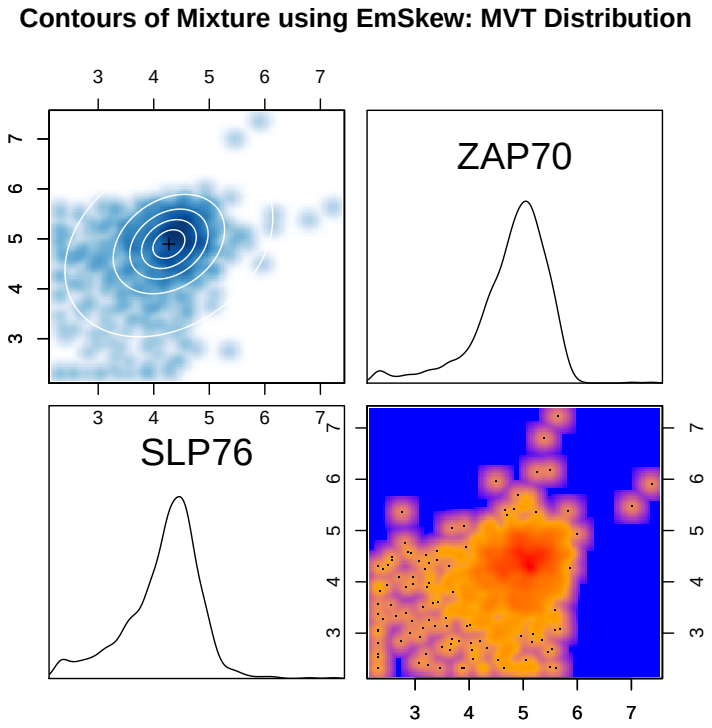

Figure 4.33: Contours for Lympho data fitted to a 1-component model of a bivariate non skewed (ellipsoidal) t-distribution via the EM algorithm using the burn-in concepts as the initialization method.

DLBCL data set

The original Diffuse Large B-cell Lymphoma (`DLBCL`) of [39], is a data set containing measurements of fluorescent intensities of multiple conjugated antibodies (Markers) stained on a sample of thousands of individual cells derived from the lymph nodes of 30 patients diagnosed with defuse large B-cell lymphoma [40]. The `DLBCL` data set for this study, is only a subset containing over 8000 cells from one patient. Each sample point from the sample used in this study, was stained with three antibodies, CD3, CD5, and CD19. This gives a data set that has three variables namely, Marker 1 (for CD3), Marker 2 (for CD5) and Marker 3 (for CD19) measured on each of the $n = 8000$ observed individual cells. In this study, we take the `DLBCL` sample data to be a three dimensional data set from a 4-component mixture distribution [11]. In order to group the data set into 4 clusters, the data set can be fit to a mixture model with $g = 4$ components that are t-distributed. Figure 4.34 shows a smooth scatter plot of the `DLBCL` data set.

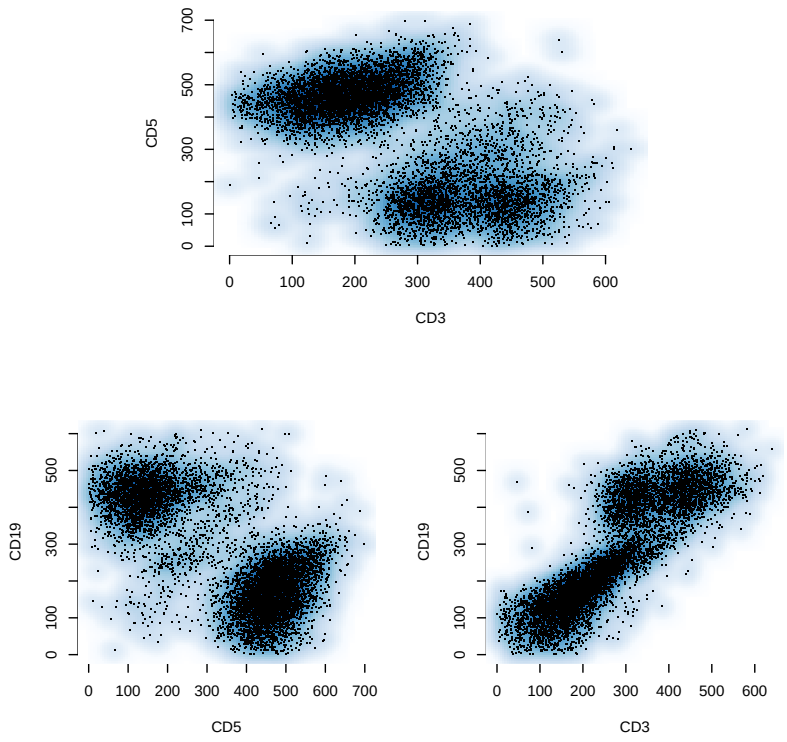

Figure 4.34: Smoothed scatter plots for the variables in the DLBCL data

We fitted the `DLBCL` data set to a 4-component mixture of multivariate t-distributions via the EM algorithm using burn-in scheme as the initialization method. Further, we fitted the `DLBCL` data set to a 4-component mixture of multivariate t-distributions via the EM algorithm using the k-means as the initialization method. A comparison of the average results based on 100 simulations is shown in Table 4.28. Based on the value of the average convergent log-likelihood values, the burn-in scheme was a better EM initialization method for the `DLBCL` data set.

	Using Burn-in scheme	Using k-means algorithm
`conv.error`	0	0
`BIC`	285997.6	286031.1
`AIC`	285843.4	285876.9
$l(\hat{\boldsymbol{\Psi}})$	-142899.7	-142916.5

Table 4.28: A comparison of the average EM algorithm output values based on 100 simulations for the burn-in scheme and k-means algorithm as initialization methods when the DLBCL data set is fit to a 4-component mixture of multivariate t-distributions.

Table 4.29 compares average estimates based on 100 simulations from the EM algorithm when burn-in scheme and k-means algorithm are used as initialization methods. The output values from burn-in scheme are compared with results from k-means since the k-means dominated both the hierarchial clustering and random start methods. From the BIC and AIC values in Table 4.28, burn-in scheme produces slightly lower values than the k-means algorithm. Hence, the estimates in Table 4.29 associated with the burn-in scheme are considered superior to those associated with the k-means methods.

par.	Parameter(par.) Estimate When EM is Initialized via k-means Algorithm	Burn-in scheme
$\hat{\boldsymbol{\tau}}$	$(.19,\ .33,\ .19,\ 0.28)^T$	$(.19,\ .32,\ .19,\ .30)^T$
$\hat{\boldsymbol{\nu}}$	$(3, 12, 4, 4)^T$	$(3, 161, 4, 4)^T$
$\hat{\boldsymbol{\mu}}_1$	$(446.18, 160.31, 449.694)^T$	$(446.23, 160.78, 449.89)^T$
$\hat{\boldsymbol{\mu}}_2$	$(142.08, 449.93, 124.09)^T$	$(138.87, 449.67, 120.97)^T$
$\hat{\boldsymbol{\mu}}_3$	$(318.88, 136.01, 411.22)^T$	$(319.16, 135.78, 411.21)^T$
$\hat{\boldsymbol{\mu}}_4$	$(241.47, 488.48, 218.63)^T$	$(239.17, 486.46, 216.36)^T$
$\hat{\boldsymbol{\Sigma}}$	$\text{diag}(1733, 3263, 1836)$	$\text{diag}(1820, 33245, 1907)$

Table 4.29: Average estimates from the EM algorithm based on 100 simulations, EM initialized via various (a) k-means algorithm (b) burn-in scheme

Note that the fitted models here assume that the variance-covariance matrices or scale matrices are equal across the four components so that

$$\hat{\boldsymbol{\Sigma}}_1 = \hat{\boldsymbol{\Sigma}}_2 = \hat{\boldsymbol{\Sigma}}_3 = \hat{\boldsymbol{\Sigma}}_4 = \hat{\boldsymbol{\Sigma}}$$

CHAPTER 5
Discussion and Conclusion

This chapter presents the concluding remarks of this study. The answers to the research questions are summarized based on the experimental results. Further work that can be taken in line with this research is suggested.

This study has presented a thorough review of the theoretical principles underlying the concept of finite mixture models. Specifically, finite mixture models featuring multivariate t-distribution have been reviewed as they are used in the clustering of continuous multivariate data sets. In this study, the usefulness of finite mixtures of multivariate t-distributions in providing a mathematical based approach in statistical modelling of continuous multivariate data sets has been highlighted. The study has stressed the importance of the t-distribution as a longer tailed alternative to the Gaussian distribution in mixture modelling, as the former is more adapted to dealing with data sets containing outliers. Further, the usefulness of the EM algorithm as a tool for determining MLE in finite mixture models has been adequately reviewed with emphasis on its application to component parameter estimation when the mixture model features multivariate t-distributions.

The performance of the EM algorithm in determining the optimal estimate of the mixture model parameter $\mathbf{\Psi}$ is well known to be heavily dependent on the choice of the initial starting point for the EM iterations. This study has reviewed three of the most common methods used in the selection of the starting point $\mathbf{\Psi}^0$, when fitting continuous multi-variable data sets to finite mixture models that feature components with multivariate t-distributions via the EM algorithm. Common initialization methods employed in the EM include; the k-means clustering algorithm, the hierarchical clustering algorithm and random start methods [10][11][21]. The theoretical principles underlying these initialization methods as well as their application in the EM process, has been studied and details presented. Further, we have investigated an alternative method called the burn-in scheme [4][5], as a means of selecting the starting point for the EM algorithm when the underlying mixture model has non-Gaussian components, in this case the t-distributions. The application of these four methods as EM initialization has been implemented primarily using the package `EMMIXskew` in the statistical software **R**.

Four simulated data sets were used in this study. Two data sets were simulated from mixture distributions with components that have multivariate

Gaussian distributions. Fitting these two data sets to mixtures of multivariate t-distributions showed that the proposed burn-in scheme competes favourably in performance compared to the k-means algorithm, which often outperformed both the hierarchical clustering algorithm and the random start method. The two simulated data sets from mixtures of Gaussian distributions, present a testing ground for the performance of the burn-in scheme in mixture modelling featuring multivariate t-distributions, for populations that may be normally or approximately normally distributed. The other two data sets were simulated from mixture models whose components have multivariate t-distributions. Unlike the data sets simulated from mixtures of Gaussian distributions, these data sets exhibited a heavier tail behavior associated with the t-distribution. The results of fitting each of these two data sets to mixtures of multivariate t-distributions show that the burn-in scheme is a competitive EM initialization method as its performance in comparison to the dominantly used methods often yielded competitive results.

Six real data sets were used in further testing of the performance of the burn-in scheme. Here, the iris data set is assumed to be from a mixture distribution featuring four multivariate t-distributions with equal variances across the components. The results from this data set show that implementing the EM process using the burn-in scheme rarely hurts as its performance compared favourably with those of the other dominant methods. In fact, on average, all the methods gave identical results. This could be due to the fact that the likelihood function associated with the iris data is not extremely multi-modal [4]. The important point emanating from this observation is that the performance of the burn-in scheme is not below that of the current dominant EM initialization methods. Similar trends were observed when fitting `ais` data to mixtures of multivariate t-distributions. In the case of the `banknote` and `faithful geyser` data sets, the burn-in scheme is observed to have outperformed all the other methods on average. This shows that the burn-in scheme could actually be a more popular EM initialization method than either the k-means clustering algorithm or the hierarchical clustering algorithm.

The burn-in scheme has shown its promising usage in the analysis of flow cytometry data sets, specifically the `Lympho` and `DLBCL` data sets. Comparing the performance of the burn-in scheme with that of the k-means initialization shows that the burn-in scheme can be adopted as the optimal EM initialization method when determining parameter estimates in distribution-based cluster analysis such as the analysis in flow cytometric analysis.

In summary, the common dominant methods of initializing the model parameters when fitting data to finite mixtures of multivariate t-distributions via the EM algorithm are; the k-means algorithm, the hierarchical clustering and random start methods. Our study of these initialization methods has demonstrated that the new burn-in scheme method can be used to effect parameter initialization in the EM algorithm with just as good a chance of

convergence to the global mode, even when the underlying mixture model features components with t-distributions. This study has demonstrated and established the following important points.

(i) The burn-in concepts of O'Hagan et al (2012) can easily be extended as an alternative initialization method to the EM algorithm when fitting data to finite mixtures of multivariate t-distributions.

(ii) When fitting data to finite mixtures of multivariate t-distributions via EM algorithm, the burn-in scheme compares favourably as an alternative initialization method to the dominant methods such as random starts, k-means and hierarchical clustering

(iii) When fitting data to finite mixtures of multivariate t-distributions via EM algorithm, the optimal initialization method can not easily be pre-determined, although the k-means and the burn-in scheme often outperforms the random methods and hierarchical clustering. In a few instances however, it was noted that the random start approach can outperform both the k-means and the burn-in scheme.

Suggested further studies in line with this research work include:

(a) constructing parameter initialization methods for the EM algorithm that compound the techniques of the burn-in scheme and those of (i) k-means algorithm (ii) hierarchical clustering, and (ii) random start methods, in the context of mixtures of multivariate t-distributions.

(b) There is a lot of interest in using mixture models that feature skew-normal distributions and skew-t distributions in distribution based mixture modeling as these models can effect better clustering solutions when the underlying data set is asymmetric or has non-elliptical groups [10][11][12]. Therefore, extending the concepts of the burn-in scheme of O'Hagan et al (2012) and the proposed initialization methods in (a) to such models would be an exciting topic of study.

Appendix

We present details of implementing the EM algorithm in the statistical software **R**. We initialize the EM algorithm via (i) k-means (ii) random start (iii) hierarchical clustering and (iv) burn scheme. We use the `iris` data set, but the methods can be applied to any data set.

```
> library(EMMIXuskew)
> library(EMMIXskew)
> library(mclust)
> data(iris)
```

We fit `iris` data to a mixture of three multivariate t-distributions via the EM algorithm. We initialize the EM algorithm using four methods:

i) Generate the initial values using k-means algorithm and fit `iris` data to a mixture model of multivariate t-distributions.

```
> iris.Fit1.init.1 <- init.mix(iris[,1:4], g=3, distr="mvt",
+ ncov=2, nkmeans=1, nrandom=0, nhclust=FALSE, maxloop=10)
> iris.Fit1.init.2 <- init.mix(iris[,1:4], g=3, distr="mvt",
+ ncov=2, nkmeans=10, nrandom=0, nhclust=FALSE, maxloop=1)
```

```
> iris.Fit1.1 <- EmSkewfit2(dat=iris[,1:4], g=3, distr="mvt",
+ init=iris.Fit1.init.1, ncov=2, itmax=100, epsilon=1e-6)
> iris.obj1.1 <- c(iris.Fit1.1$loglik, iris.Fit1.1$error,
+ iris.Fit1.1$aic, iris.Fit1.1$bic)
> names(iris.obj1.1)<-c("conv.loglik","conv.error","AIC","BIC")
> iris.obj1.1
```

```
conv.loglik  conv.error         AIC         BIC
  -344.0613      0.0000    730.1225    793.3459
```

```
> iris.Fit1.2 <- EmSkewfit2(dat=iris[,1:4], g=3, distr="mvt",
+ init=iris.Fit1.init.2, ncov=2, itmax=100, epsilon=1e-9)
> iris.obj1.2 <- c(iris.Fit1.2$loglik,iris.Fit1.2$error,
+ iris.Fit1.2$aic,iris.Fit1.2$bic)
> names(iris.obj1.2)<-c("conv.loglik","conv.error","AIC","BIC")
> iris.obj1.2
```

```
conv.loglik  conv.error         AIC         BIC
  -344.0611      1.0000    730.1223    793.3456
```

The object `iris.obj1.1` contains selected output values of the EM algorithm initialized using a single k-means and `iris.obj1.2` contains selected output values of the EM algorithm initialized using 10 k-means partitions. To see all the EM algorithm output values (ML-estimates of $\mathbf{\Psi}$), the commands

`print(iris.Fit1.1)` and `print(iris.Fit1.2)`

maybe issued in **R** console. This gives all the estimated model parameters.

ii) Initialize the EM algorithm using the random start method and fit the `iris` data to a mixture model of multivariate t-distributions.

```
> iris.Fit2.init.1 <- init.mix(iris[,1:4], g=3, distr="mvt",
+ ncov=2, nrandom=1,nkmeans=0, nhclust=FALSE, maxloop=10)
> #iris.Fit2.init.1
> iris.Fit2.init.2 <- init.mix(iris[,1:4], g=3, distr="mvt",
+ ncov=2, nrandom=10, nkmeans=0, nhclust=FALSE, maxloop=1)
> #iris.Fit2.init.2

> iris.Fit2.1 <- EmSkewfit2(dat=iris[,1:4], g=3, distr="mvt",
+ init=iris.Fit2.init.1, ncov=2, itmax=100, epsilon=1e-6)
> iris.obj2.1 <- c(iris.Fit2.1$loglik,iris.Fit2.1$error,
+ iris.Fit2.1$aic,iris.Fit2.1$bic)
> names(iris.obj2.1)<-c("conv.loglik","conv.error","AIC","BIC")
> iris.obj2.1

conv.loglik  conv.error         AIC         BIC
  -344.0612      0.0000    730.1223    793.3457

> iris.Fit2.2 <- EmSkewfit2(dat=iris[,1:4], g=3, distr="mvt",
+ init=iris.Fit2.init.2, ncov=2, itmax=100, epsilon=1e-9)
> iris.obj2.2 <- c(iris.Fit2.2$loglik,iris.Fit2.2$error,
+ iris.Fit2.2$aic, iris.Fit2.2$bic)
> names(iris.obj2.2)<-c("conv.loglik","conv.error","AIC","BIC")
> iris.obj2.2

conv.loglik  conv.error         AIC         BIC
  -344.0611      0.0000    730.1223    793.3456
```

The object `iris.obj2.1` contains outputs values from the EM using one random start and `iris.obj2.2` contains output values from the EM algorithm initialized using 10 random starts, and selecting the best performing.

iii) Initialize the EM algorithm using the hierarchical clustering method and fit the `iris` data to a mixture model of three multivariate t-distributions.

```
> iris.Fit3.init <- init.mix(iris[,1:4], g=3, distr="mvt",
+ ncov=2, nrandom=0, nkmeans=0, nhclust=TRUE)
> #iris.Fit3.init

> iris.Fit3 <- EmSkewfit2(dat=iris[,1:4], g=3, distr="mvt",
+  init=iris.Fit3.init,ncov=2, itmax=100, epsilon=1e-6)
> iris.obj3 <- c(iris.Fit3$loglik,iris.Fit3$error,
+ iris.Fit3$aic,iris.Fit3$bic)
> names(iris.obj3)<-c("conv.loglik","conv.error","AIC","BIC")
> iris.obj3
```

iv) Initialize the EM algorithm using the burn-in concepts and fit the `iris` data to a mixture model of three multivariate t-distributions.

Generate the clusters so that we get the required $\boldsymbol{Z}^{(o)}$ matrices. We take $b = 5$ so that we have:

```
> Irisclust <- list(Irisclust1,Irisclust2,Irisclust3,
+ Irisclust4,Irisclust5,Irisclust6,Irisclust7,Irisclust8,
+ Irisclust9,Irisclust10,Irisclust11,Irisclust12,Irisclust13,
+ Irisclust14,Irisclust15,Irisclust16,Irisclust17,Irisclust18,
+ Irisclust19,Irisclust20,Irisclust21,Irisclust22,Irisclust23,
+ Irisclust24,Irisclust25,Irisclust26,Irisclust27,Irisclust28,
+ Irisclust29,Irisclust30,Irisclust31,Irisclust32)

> Fit4.single <- function(data,g,clust,distr="mvt",ncov=2,
+ itmax,epsilon,initloop,debug=FALSE){
+ obj1.data <- EmSkewfit1(data, g, clust, distr="mvt",
+ ncov, itmax, epsilon,initloop)
+ #labz <- paste(c("cluster:"), obj$loglik, sep="")
+ #print(labz)
+ print(obj1.data$loglik)
+ }
> Fit4.clust.1 <- function(clust){
+ obj2.iris <- Fit4.single(iris[,1:4],clust, g=3,distr="mvt",
+ ncov=2,itmax=2,epsilon=1e-6,initloop=2,debug=FALSE)
+ }
> Fit4.1 <- function(data,g,dataclust,distr="mvt",ncov=2,
+ itmax,epsilon,initloop,b,debug=FALSE){
+ objdata.AllLogliks <- lapply(dataclust,Fit4.clust.1)
+ print(objdata.AllLogliks)
+ }

> Fit4.single(iris[,1:4], g=3,clust=Irisclust1,distr="mvt",
+ ncov=2,itmax=2,epsilon=1e-6,initloop=2)
> Fit4.clust.1(clust=Irisclust1)
> iris.AllLoglikes.burn.1 <- Fit4.1(data=iris[,1:4],g=3,
+ itmax=2,dataclust=Irisclust,distr="mvt",ncov=2,
```

```
+ epsilon=1e-6,initloop=2,b=16,debug=FALSE)
> iris.AllLoglikes.vector.1 <- c(do.call("cbind",
+ iris.AllLoglikes.burn.1))
> names(iris.AllLoglikes.vector.1) <- c("Irisclust1",
+ "Irisclust2","Irisclust3","Irisclust4","Irisclust5",
+ "Irisclust6","Irisclust7","Irisclust8","Irisclust9",
+ "Irisclust10","Irisclust11","Irisclust12","Irisclust13",
+ "Irisclust14","Irisclust15","Irisclust16","Irisclust17",
+ "Irisclust18","Irisclust19","Irisclust20","Irisclust21",
+ "Irisclust22","Irisclust23","Irisclust24","Irisclust25",
+ "Irisclust26","Irisclust27","Irisclust28","Irisclust29",
+ "Irisclust30","Irisclust31","Irisclust32")
> sort(iris.AllLoglikes.vector.1,decreasing=TRUE) -> d
> sorted.iris.AllLoglikes.vector.1 <- d
> b <- length(sorted.iris.AllLoglikes.vector.1)/2; b
> dA <- sorted.iris.AllLoglikes.vector.1[1:b]
> sorted.iris.HalfLoglikes.vector.1 <- dA
> Irisclust.2 <- list(Irisclust1,Irisclust2,Irisclust3,
+ Irisclust4,Irisclust5,Irisclust6,Irisclust7,Irisclust8,
+ Irisclust9,Irisclust10,Irisclust11,Irisclust12,Irisclust25,
+ Irisclust26,Irisclust31,Irisclust32)
> Fit4.clust.2 <- function(clust){
+ obj2.0iris <- Fit4.single(iris[,1:4],clust, g=3,distr="mvt",
+ ncov=2,itmax=4,epsilon=1e-6,initloop=2,debug=FALSE)
+ }
> Fit4.2 <- function(data,g,dataclust,distr="mvt",ncov=2,itmax,
+ epsilon,initloop,b,debug=FALSE){
+ objdata.AllLogliks.2 <- lapply(dataclust,
+ Fit4.clust.2)
+ print(objdata.AllLogliks.2)
+ }
> iris.AllLoglikes.burn.2 <- Fit4.2(iris[,1:4],g=3,distr="mvt",
+ dataclust=Irisclust.2,ncov=2,itmax=4,
+ epsilon=1e-6,initloop=2,b=8)
> iris.AllLoglikes.vector.2 <- c(do.call("cbind",
+ iris.AllLoglikes.burn.2))
> names(iris.AllLoglikes.vector.2) <- c("Irisclust1",
+ "Irisclust2","Irisclust3","Irisclust4","Irisclust5",
+ "Irisclust6","Irisclust7","Irisclust8","Irisclust9",
+ "Irisclust10","Irisclust11","Irisclust12","Irisclust25",
+ "Irisclust26","Irisclust31","Irisclust32")
> sort(iris.AllLoglikes.vector.2,decreasing=TRUE) <- dA1
> Sorted.iris.AllLoglikes.vector.2 <- dA1
> b.1 <- length(sorted.iris.AllLoglikes.vector.1)/4; b.1
> Sorted.iris.AllLoglikes.vector.2[1:b.1] <- dH1
> Sorted.iris.HalfLoglikes.vector.2 <- dH1
> Sorted.iris.HalfLoglikes.vector.2
> Irisclust.3 <- list(Irisclust1,Irisclust2,Irisclust3,
+ Irisclust4,Irisclust5,Irisclust6,Irisclust7,Irisclust10)
```

```
> Fit4.clust.3 <- function(clust){
+ obj2.1iris <- Fit4.single(iris[,1:4],clust, g=3,distr="mvt",
+ ncov=2,itmax=8,epsilon=1e-6,initloop=2,debug=FALSE)
+ }
> Fit4.3 <- function(data,g,dataclust,distr="mvt",ncov=2,itmax,
+ epsilon,initloop,b,debug=FALSE){
+ objdata.AllLogliks.3 <- lapply(dataclust,Fit4.clust.3)
+ print(objdata.AllLogliks.3)
+ }
> iris.AllLoglikes.burn.3 <- Fit4.3(iris[,1:4],g=3,distr="mvt",
+ dataclust=Irisclust.3,ncov=2,itmax=8,
+ epsilon=1e-6,initloop=2,b=4)
> iris.AllLoglikes.vector.3 <- c(do.call("cbind",
+ iris.AllLoglikes.burn.3))
> names(iris.AllLoglikes.vector.3) <- c("Irisclust1",
+ "Irisclust2","Irisclust3","Irisclust4","Irisclust5",
+ "Irisclust6","Irisclust7","Irisclust10")
> dA3 <- sort(iris.AllLoglikes.vector.3,decreasing=TRUE)
> Sorted.iris.AllLoglikes.vector.3 <- dA3
> b.2 <- length(sorted.iris.AllLoglikes.vector.1)/8; b.2
> dH3 <-  Sorted.iris.AllLoglikes.vector.3[1:b.2]
> Sorted.iris.HalfLoglikes.vector.3 <-  dH3
> Irisclust.4 <- list(Irisclust1,Irisclust2,
+ Irisclust4,Irisclust6)
> Fit4.clust.4 <- function(clust){
+ obj2.2iris <- Fit4.single(iris[,1:4],clust, g=3,distr="mvt",
+ ncov=2,itmax=10,epsilon=1e-6,initloop=2,debug=FALSE)
+ }
> Fit4.4 <- function(data,g,dataclust,distr="mvt",ncov=2,itmax,
+ epsilon,initloop,b,debug=FALSE){
+ objdata.AllLogliks.4 <- lapply(dataclust,Fit4.clust.4)
+ print(objdata.AllLogliks.4)
+ }
> iris.AllLoglikes.burn.4 <- Fit4.4(iris[,1:4],g=3,itmax=10,
+ dataclust=Irisclust.4,distr="mvt",ncov=2,
+ epsilon=1e-6,initloop=2,b=2)
> iris.AllLoglikes.burn.4
> length(iris.AllLoglikes.burn.4)
> iris.AllLoglikes.vector.4 <- c(do.call("cbind",
+ iris.AllLoglikes.burn.4))
> names(iris.AllLoglikes.vector.4) <- c("Irisclust1",
+ "Irisclust2","Irisclust4","Irisclust6")
> dA4 <- sort(iris.AllLoglikes.vector.4,decreasing=TRUE)
> Sorted.iris.AllLoglikes.vector.4 <- dA4
> Sorted.iris.AllLoglikes.vector.4
> b.3 <- length(sorted.iris.AllLoglikes.vector.1)/16; b.3
> dH4 <- Sorted.iris.AllLoglikes.vector.4[1:b.3]
> Sorted.iris.HalfLoglikes.vector.4 <- dH4
> Sorted.iris.HalfLoglikes.vector.4
```

```
> Irisclust.5 <- list(Irisclust1,Irisclust6)
> Fit4.clust.5 <- function(clust){
+ obj2.3iris <- Fit4.single(iris[,1:4],clust, g=3,distr="mvt",
+ ncov=2,itmax=12,epsilon=1e-6,initloop=2,debug=FALSE)
+ }
> Fit4.5 <- function(data,g,dataclust,distr="mvt",ncov=2,itmax,
+ epsilon,initloop,b,debug=FALSE){
+ objdata.AllLogliks.5 <- lapply(dataclust,Fit4.clust.5)
+ print(objdata.AllLogliks.5)
+ }
> iris.AllLoglikes.burn.5 <- Fit4.5(iris[,1:4],g=3,distr="mvt",
+ dataclust=Irisclust.5,ncov=2,itmax=12,
+ epsilon=1e-6,initloop=2,b=1)
> iris.AllLoglikes.burn.5
> length(iris.AllLoglikes.burn.5)
> iris.AllLoglikes.vector.5 <- c(do.call("cbind",
+ iris.AllLoglikes.burn.5))
> iris.AllLoglikes.vector.5
> names(iris.AllLoglikes.vector.5) <- c("Irisclust1",
+ "Irisclust6")
> iris.AllLoglikes.vector.5
> dA5 <- sort(iris.AllLoglikes.vector.5,decreasing=TRUE)
> Sorted.iris.AllLoglikes.vector.5 <-  dA5
> Sorted.iris.AllLoglikes.vector.5
> b.4 <- length(sorted.iris.AllLoglikes.vector.1)/32; b.4
> dH5 <- Sorted.iris.AllLoglikes.vector.5[1]
> Sorted.iris.HalfLoglikes.vector.5 <- dH5
> Sorted.iris.HalfLoglikes.vector.5
> dHF <- Sorted.iris.HalfLoglikes.vector.5
> Optimal.cluster.loglikelohood <- dHF
> Optimal.cluster.loglikelohood
> iris.optimal_Z <- Irisclust1
> iris.optimal_Z

> iris.optimal_init <- initEmmix(iris[,1:4], g=3,
+ distr="mvt",clust=iris.optimal_Z,ncov=2,
+ maxloop=12)
> iris.Fit4 <- EmSkewfit2(dat=iris[,1:4], g=3,
+ distr="mvt",init=iris.optimal_init,ncov=2,
+ itmax=100, epsilon=1e-6)
> iris.obj4 <- c(iris.Fit4$loglik,iris.Fit4$error,
+ iris.Fit4$aic,iris.Fit4$bic)
> names(iris.obj4)<-c("conv.log-lik","conv.error","AIC","BIC")
> iris.obj4

> print(iris.Fit1.1); print(iris.Fit1.2); print(iris.Fit3);
> print(iris.Fit2.1); print(iris.Fit2.2); print(iris.Fit4)
```

References

[1] McLachlan, G.J. and Krishnan, T. (1997). The EM Algorithm and Extensions. Wiley Interscience, New York.

[2] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, Series B, 39, 1-38.

[3] Peel, D. and McLachlan, G.J. (2000). Robust mixture modelling using the t-distribution. *Statistics and Computing*, 10(4), 339-348.

[4] O'Hagan, A., Murphy, T.B. and Gormley I.C., (2012). Computational aspects of fitting mixture models via the expectation-maximisation algorithm. *Computational Statistics and Data Analysis*, 56(12), 3843–3864.

[5] 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(1), 561–575.

[6] Dimitris, K. and Evdokia, X. (2003). Choosing Initial Values for The EM Algorithm for Finite Mixtures. *Computational Statistics and Data Analysis*, 41, 577–590.

[7] Andrews J.L., McNicholas P.D., and Subedi S. (2011). Model-based classification via mixtures of multivariate t-distributions. *Computational Statistics and Data Analysis*, 55(1), 520–529.

[8] Liu C. (1997). ML estimation of the multivariate t-distribution and the EM algorithm. *Journal of Multivariate Analysis*, 63(2), 296–312.

[9] 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*, 355–368.

[10] Mclachlan, G.J., Peel, D., Basford, K. and Adams, P. (1999). The EMMIX Software for the Fitting of Mixtures of Normal and t-components. *Journal of Statistical Software*, volume 4.

[11] Lee, S. and McLachlan, G.J. (2013). EMMIXuskew: An R package for fitting mixtures of multivariate skew t-distributions via the EM algorithm. *Journal of Statistical Software*, 55(12), 1-22.

[12] Lee, S.X. and McLachlan, G.J. (2011). On the fitting of mixtures of multivariate skew t-distributions via the EM algorithm. *arXiv preprint arXiv:1109.4706*

[13] Jeff Wu. C. F. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1), 95–103.

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

[15] Meng, X. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2), 267–278.

[16] Liu, C. and Rubin, D. B. (1994). The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. *Biometrika*, 81(4), 633–648.

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

[18] Vermunt, J. and Madison, J. (2005). Technical Guide for Latent GOLD 4.0. Basic and Advanced. Statistical Inovations Inc, Belmont Massachusetts.

[19] Liu, C. and Rubin, D.B. (1995). ML estimation of the t distribution using EM and its extensions, ECM and ECME. *Statistica Sinica*, 5(4), 19–39.

[20] Davis, P. J. (1965). Gamma function and related functions. In Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables (Edited by Abramowitz M. and Stegun, I. A.)*National Bureau of Standards, Applied Mathematics Series*, 55(1), 253–293.

[21] Wang, K., McLachlan, G.J., Ng, S.K. and Peel, D.(2012). EMMIX-skew: EM Algorithm for Mixtures of Multivariate skew Normal/t-distributions. **R** Code version 1.0-12. *http://www.maths.uq.edu.au/gjm/mix/soft/EMMIX-skew.*

[22] Lee, S. and McLachlan, G.J. (2014). Finite Mixtures of Multivariate Skew t-Distributions: Some Recent and New Results. *Statistics and Computing*, 24(2), 181–202.

[23] Kotz, S. and Nadarajah, S. (2004). Multivariate t Distributions and Their Applications. Cambridge University Press, Cambridge.

[24] McNeil, A. J. (2006). Multivariate t-Distributions and Their Applications. *Journal of the American Statistical Association*, 101(473), 390-391.

[25] Ng, S.K., Krishnan, T. and Mclachlan, G.J. (2012). The EM algorithm. In: Gentle J., Hãrdle, W., Mori, Y. (eds), Handbook of Computational Statistics: Concepts and Methods. Springer, Berlin.

[26] Chen, J., Ching R.K.H. and Lin, Y. (2004). An Extended Study of the k-Means Algorithm for Data Clustering and Its Applications. *The Journal of the Operational Research Society*, 55(9), 976–987.

[27] Hartigan, J. A. and Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the royal statistical society. Series C (Applied Statistics)*, 28(1), 100–108.

[28] Hamerly, G. and Elkan, C. (2002). Alternatives to the k-means algorithm that find better clusterings. *Proceedings of the eleventh international conference on Information and knowledge management (CIKM)*, 600–607.

[29] Forgy, E.W. (1965).Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *BiometriKa*, 21(3), 768–769.

[30] MacKay, D.J.C. (2003). An Example Inference Task Clustering: *Information Theory, Inference and Learning Algorithms*, Chapter 20, 284–292. Cambridge University Press.

[31] Bradley, P.S. and Fayyad, U.M. (1998). Refining Initial Points for *k*-Means Clustering. *Proceedings of the Fifteenth International Conference on Machine Learning.* 91–99.

[32] Inaba, M., Katoh, N. and Imai, H. (1994). Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. *Proceedings of 10th ACM Symposium on Computational Geometry*, 1(1), 332–339.

[33] Defays, D. (1977). An efficient algorithm for a complete-link method. *The Computer Journal*, 22(4), 364–366.

[34] Rokach, L. and Maimon, O. (2005). *Data mining and knowledge discovery handbook.* Springer, Berlin.

[35] McLachlan, G.J. and Peel, D. (2000). Finite Mixture Models. Wiley, New York.

[36] Fisher, R., (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(1), 179–188.

[37] Weisberg, S., (2005). Applied Linear Regression 3rd edition, New York: Wiley Section 6.4.

[38] Flury, B. and Riedwyl, H., (1988). Multivariate Statistics: A practical approach. *London: Chapman & Hall, Tables 1.1 and 1.2*, pages 5–8.

[39] Spidlen J., Breuer K., Rosenberg C., Kotecha N. and Brinkman R.R. (2012). FlowRepository - A Resource of Annotated Flow Cytometry Datasets Associated with Peer-reviewed Publications. *Cytometry A*, 81(9), 727–731.

[40] Aghaeepour, N., Finak, G., The FlowCAP Consortium. et al. (2013). Critical assessment of automated flow cytometry analysis techniques. *Nature Methods*, 10(1), 228–238.

[41] Azzalini, A. and Bowman, A. W. (1990). A look at some data on the Old Faithful geyser. Applied Statistics. *Journal of Royal Statistical Society*, Series C, 39(1), 357–365.

[42] Maier L.M., Anderson D.E., De Jager P.L., Wicker L., Hafler D.A. (2007). Allelic variant in CTLA4 alters T cell phosphorylation patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 104(47), 18607–18612.

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

[44] R Core Team (2015). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.