On Clustering: Mixture Model Averaging with the Generalized Hyperbolic Distribution

ON CLUSTERING: MIXTURE MODEL AVERAGING WITH THE GENERALIZED HYPERBOLIC DISTRIBUTION

BY

SARAH RICCIUTI, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MATHEMATICS & STATISTICS

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Master of Science (2017) McMaster University
(Mathematics & Statistics) Hamilton, Ontario, Canada

TITLE:	On Clustering: Mixture Model Averaging with the Generalized Hyperbolic Distribution
AUTHOR:	Sarah Ricciuti B.Sc., (Actuarial and Financial Math) McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Paul D. McNicholas
NUMBER OF PAGES:	viii, 42

To my parents, Sue and Rick Ricciuti, and in loving memory of my grandparents,

Katherine and Frank Mayer

Abstract

Cluster analysis is commonly described as the classification of unlabeled observations into groups such that they are more similar to one another than to observations in other groups. Model-based clustering assumes that the data arise from a statistical (mixture) model and typically a group of many models are fit to the data, from which the 'best' model is selected by a model selection criterion (often the BIC in mixture model applications). This chosen model is then the only model that is used for making inferences on the data. Although this is common practice, proceeding in this way ignores a large component of model selection uncertainty, especially for situations where the difference between the model selection criterion for two competing models is relatively insignificant. For this reason, recent interest has been placed on selecting a subset of models that are close to the selected best model and using a weighted averaging approach to incorporate information from multiple models in this set. Model averaging is not a novel approach, yet its presence in a clustering framework is minimal. Here, we use Occam's window to select a subset of models eligible for two types of averaging techniques: averaging *a posteriori* probabilities, and direct averaging of model parameters. The efficacy of these model-based averaging approaches is demonstrated for a family of generalized hyperbolic mixture models using real and simulated data.

Acknowledgements

First and foremost, I would like to express my sincere gratitude for my supervisor Dr. Paul McNicholas, for his guidance and continuous support throughout my studies. I feel very fortunate to have had the opportunity to work with him, and truly appreciate the dedication and kindness he shows towards his graduate students.

Thank you to Dr. Brian Franczak for always being open to my questions when I first started out as an undergraduate research assistant on the McNicholas team. Coding is a steep learning curve.

I would also like to thank my family. Thank you to my grandparents Katherine and Frank Mayer for their unconditional love and for always being there for me. I am the person I am today because of them. Thank you to my parents Sue and Rick Ricciuti for providing endless encouragement throughout my studies, and constantly inspiring me to achieve my goals. Thank you to my brother Paul for setting an excellent academic example. And a very special thank you to my Golden Retriever Maddie for keeping me company during the late nights of working.

Additionally, I would like to acknowledge the financial support I have received from the NSERC Discovery Grant and the Canadian Research Chairs program.

Contents

Abstract **iv**

Acknowledgements **v**

1 Introduction **1**

2 Background **3**

2.1 Clustering 3

2.2 Finite Mixture Models and Model-Based Clustering 4

2.3 Mixture of Generalized Hyperbolic Distributions 5

2.3.1 Generalized Inverse Gaussian Distribution 5

2.3.2 Generalized Hyperbolic Distribution 6

2.3.3 Parameter Estimation 7

2.3.4 Convergence 7

2.3.5 Predicted Classifications 8

2.4 Model Selection 9

2.5 Performance Assessment 9

2.6 *A Posteriori* Merging of Components 10

2.7 Bayesian Model Averaging 12

2.8 Occam's Window 13

2.9 Approximation with the BIC 14

3 Methodology 16

3.1 Merging Mixture Components 16

3.2 Merging Process Example 17

3.3 Averaging *A Posteriori* Probabilities 19

3.4 Direct Model Averaging 20

3.5 Label Switching 21

4 Analyses 22

4.1 Performance and Evaluation 22

4.1.1 Purpose 22

4.1.2 Generation of Clustering Models 23

4.2 Simulation Scenarios 23

4.2.1 Scenario I: Easy Clustering Problem 24

4.2.2 Scenario II: Hard Clustering Problem 25

4.3 Applications to Datasets 29

4.3.1 Yeast 29

4.3.2 AIS 32

4.3.3 Examples with only one model in Occam's window: crabs data, and wine data 34

5 Conclusions 36

Bibliography 38

List of Figures

4.1 Pairs plot for the simulated data in Scenario I. 24

4.2 Pairs plot for the simulated data in Scenario II. 26

4.3 Pairs plot for the yeast data. 30

4.4 Pairs plot for the AIS data. 33

Chapter 1

Introduction

Model averaging presents a mechanism for accounting for the fact that a model selection criterion, often the Bayesian information criterion (BIC; Schwarz, 1978), may not select the model with the correct number of components or attain the most optimal classification results (Biernacki *et al.*, 2000). Additionally, models that fit the data to a similar degree may lead to very different inferences. Hoeting *et al.* (1999) provide an example in a hazards regression context for the analysis of esophagus cancer patients; challenges arise for designing intervention strategies and predicting survival time for future patients when two models fit the data well, yet each lead to very different parameter estimates or life expectancy predictions.

The use of mixture models for clustering has attracted much attention over the past decade or so. Traditionally, Gaussian components were used but, more recently, non-Gaussian components have gained popularity (see McNicholas, 2016b, for a recent review). Wei and McNicholas (2015) applied two averaging techniques, *a posteriori* model averaging and direct averaging of model parameters, to mixtures of the Gaussian parsimonious clustering models (GPCM; Celeux and Govaert, 1995), and found that these averaging approaches successfully enhanced clustering performance for several data sets available in R as well as simulated data sets. The major difference between the two methods is that the direct averaging of model parameters method produces a single interpretable model, whereas the *a posteriori* model averaging method does not. Here, we extend these averaging methods to mixtures of generalized hyperbolic distributions. In comparison to the Gaussian distribution, the generalized hyperbolic distribution is very flexible, and it would be expected that merging components would be required more often when clustering with Gaussian components than when clustering with generalized hyperbolic components. The generalized hyperbolic distribution has the capability of modelling a skewed cluster that may require several Gaussian components to model.

Chapter 2

Background

2.1 Clustering

Cluster analysis is frequently defined as the classification of unlabelled observations into groups such that they are more similar to one another than to observations in other groups. However, this definition can be troublesome, because at the extreme case each observation would be placed in its own cluster. In an alternative definition, a cluster should consist of a set of observations that diffuse from a mode (McNicholas, 2016a). McNicholas (2016a) states that if an appropriate density is used to cluster the data, it will have the required flexibility and parametrization such that each cluster is a unimodal component within the mixture model. In a true clustering problem, the group membership of all of the observation points are not known. Determining the number and nature of the clusters as well as accurately labelling each observation point are problems that the data analyst will encounter when clustering. Approaches to evaluate model selection and the performance of different clustering approaches will be described in later sections. Clustering has a wide range of applications and many examples are available, two of which include: McNicholas and Subedi (2012) for clustering gene expression time course data to discovering genes that have similar functions, and Franczak *et al.* (2015) for clustering consumer liking studies of different bread types.

2.2 Finite Mixture Models and Model-Based Clustering

Finite mixture models assume that a population is a convex combination of a finite number of densities. In other words, it is assumed that p-dimensional data $\mathbf{x}_1, ..., \mathbf{x}_n$ have arisen from an underlying mixture where there are a finite number of distributions. The probability density function of a mixture model can be written,

$$f(\mathbf{x} \mid \boldsymbol{\Theta}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x} \mid \boldsymbol{\theta}_g), \tag{2.1}$$

where $f_g(\mathbf{x} \mid \boldsymbol{\theta}_g)$ is the probability density function of the gth component, $\boldsymbol{\Theta} = (\pi_1, ...\pi_G, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_G)$ is the vector of parameters, π_g are the mixing proportions, $\pi_g > 0$ and $\sum_{g=1}^{G} \pi_g = 1$. McNicholas (2016a) defines model-based clustering as:

"the process of clustering via a statistical mixture model"

Wolfe (1965) was the first to use finite mixture models for clustering, and it has become one of the most popular methods for clustering in the literature. Often in model-based clustering applications, each mixture component is taken to correspond to a cluster in a one-to-one relationship. However, this is not always the case and will be discussed later on in greater detail.

2.3 Mixture of Generalized Hyperbolic Distributions

2.3.1 Generalized Inverse Gaussian Distribution

The Generalized Inverse Gaussian distribution (GIG) was first introduced by Good (1953), and has since been discussed in great detail by Barndorff-Nielsen (1997), Blæsild (1978), Barndorff-Nielsen and Halgreen (1977), and Jørgensen (1982). Using the same notation as McNicholas (2016a), let W denote a random variable following a generalized inverse Gaussian distribution. The probability density function of W can be written,

$$q(w \mid a, b, \lambda) = \frac{(a/b)^{\lambda/2} w^{\lambda-1}}{2K_\lambda(\sqrt{ab})} \exp\left\{-\frac{aw + b/w}{2}\right\}, \tag{2.2}$$

where $w > 0$, K_λ is the modified Bessel function of the third kind, $\lambda \in \mathbb{R}$ is an index parameter, and a, $b \in \mathbb{R}^+$. The gamma distribution (where $b = 0$, $\lambda > 0$) and the inverse Gaussian distribution (where $\lambda = -1/2$) are two popular special cases of the GIG distribution.

2.3.2 Generalized Hyperbolic Distribution

The probability density function of the generalized hyperbolic distribution introduced by McNeil *et al.* (2005) is,

$$\begin{aligned}f(\mathbf{x} \mid \boldsymbol{\theta}) = &\left[\frac{\chi + \delta(\mathbf{x}, \boldsymbol{\mu} \mid \boldsymbol{\Delta})}{\psi + \boldsymbol{\gamma}'\boldsymbol{\Delta}^{-1}\boldsymbol{\gamma}}\right]^{(\lambda - p/2)/2} \\ &\times \frac{[\psi/\chi]^{\lambda/2} K_{\lambda - p/2}(\sqrt{[\psi + \boldsymbol{\gamma}'\boldsymbol{\Delta}^{-1}\boldsymbol{\gamma}][\chi + \delta(\mathbf{x}, \boldsymbol{\mu}|\boldsymbol{\Delta})]})}{(2\pi)^{p/2}|\boldsymbol{\Delta}|^{1/2} K_{\lambda}(\sqrt{\chi\psi})\exp\{-(\boldsymbol{\mu} - \mathbf{x})'\boldsymbol{\Delta}^{-1}\boldsymbol{\gamma}\}},\end{aligned} \tag{2.3}$$

where p is the number of variables, $\boldsymbol{\mu}$ is the location parameter, $\boldsymbol{\gamma}$ is the skewness parameter, $\boldsymbol{\Delta}$ is a $p \times p$ scale matrix (such that $|\boldsymbol{\Delta}| = 1$), λ is the index parameter, ψ and χ are concentration parameters, $\delta(\mathbf{x}, \boldsymbol{\mu}|\boldsymbol{\Delta}) = (\boldsymbol{\mu} - \mathbf{x})'\boldsymbol{\Delta}^{-1}(\boldsymbol{\mu} - \mathbf{x})$ (the squared Mahalanobis distance between $\mathbf{x}$ and $\boldsymbol{\mu}$), K_λ is the modified Bessel function of the third kind, and the vector of parameters is $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Delta}, \boldsymbol{\gamma}, \lambda, \chi, \psi)$.

Browne and McNicholas (2015) propose an alternative parametrization of the generalized hyperbolic distribution from (2.3) for the purpose of classification and clustering applications by combining a random variable following a GIG distribution and a latent multivariate Gaussian random variable, and posing constraints on the parameters (see Browne and McNicholas (2015) for details). The density of the parametrization by Browne and McNicholas (2015) can be described by,

$$\begin{aligned}f_H(\mathbf{x} \mid \boldsymbol{\theta}) = &\left[\frac{\omega + \delta(\mathbf{x}, \boldsymbol{\mu} \mid \boldsymbol{\Sigma})}{\omega + \boldsymbol{\alpha}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}}\right]^{(\lambda - p/2)/2} \\ &\times \frac{K_{\lambda - p/2}(\sqrt{[\omega + \boldsymbol{\alpha}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}][\omega + \delta(\mathbf{x}, \boldsymbol{\mu}|\boldsymbol{\Sigma})]})}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2} K_{\lambda}(\omega)\exp\{-(\mathbf{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}\}},\end{aligned} \tag{2.4}$$

where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\alpha}, \omega, \lambda)$ is the vector of parameters, $\boldsymbol{\mu}$ denotes the location parameter, $\boldsymbol{\Sigma}$ is the scale matrix, $\boldsymbol{\alpha}$ denotes the skewness parameter, ω is the concentration parameter, and λ is the index parameter.

2.3.3 Parameter Estimation

The expectation maximization (EM) algorithm (Dempster *et al.*, 1977) is used to carry out parameter estimation for the mixture of generalized hyperbolic distributions. The EM algorithm is an iterative algorithm that is often used when trying to estimate model parameters in the presence of missing or incomplete data. The EM algorithm alternates between two steps: the E-step and the M-step. In the E-step, the expected value of the complete-data log likelihood is computed and in the M-step, the expected value of the complete-data log-likelihood is maximized to get the updates for parameter estimates. See Browne and McNicholas (2015) for details on the calculations for each of these steps.

2.3.4 Convergence

Aitken's acceleration criterion (Aitken, 1926) can be used to determine when the EM algorithm has converged. Aitken's acceleration estimates the asymptotic maximum of the log-likelihood at each iteration of the EM algorithm and at iteration k can be written,

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

where $l^{(k)}$ denotes the log-likelihood at iteration k. The asymptotic estimate of the log-likelihood at iteration $k + 1$ can be computed via

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

McNicholas *et al.* (2010) considers the algorithm to have attained convergence when,

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

where ϵ is a small value.

2.3.5 Predicted Classifications

After the parameters have been estimated, the *a posteriori* probabilities (expectations) are calculated via

$$\hat{z}_{ig} := \frac{\hat{\pi}_g f_H(\mathbf{x}_i \mid \hat{\boldsymbol{\theta}}_g)}{\sum_{h=1}^{G} \hat{\pi}_h f_H(\mathbf{x}_i \mid \hat{\boldsymbol{\theta}}_h)}, \tag{2.8}$$

for $i = 1, \ldots, n$ and $g = 1, \ldots, G$. These $\hat{z}_{ig}$ represent each observation's probability of belonging to each component, and can also be referred to as "soft" classifications. Soft classifications can be useful for analyzing and comparing the results from different clustering methods, but often the $\hat{z}_{ig}$ are "hardened" to 0 or 1 based on maximum *a posteriori* probabilities, ie.,

$$\text{MAP}\{\hat{z}_{ig}\} = \begin{cases} 1 & \text{if } g = \text{argmax}_h\{\hat{z}_{ih}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

2.4 Model Selection

We now discuss a criterion that is often used to select an appropriate model. Ever since its use by Dasgupta and Raftery (1998), the BIC has become one of the most popular approaches for mixture model selection. The BIC can be written,

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

where $\hat{\boldsymbol{\vartheta}}$ is the maximum likelihood estimate of $\boldsymbol{\vartheta}$, $l(\mathbf{x}, \hat{\boldsymbol{\vartheta}})$ is the maximized log-likelihood, ρ is the number of free parameters, and n is the number of observations.

2.5 Performance Assessment

The adjusted Rand index (ARI; Hubert and Arabie, 1985) can be used to compare partitions for evaluating classification performance in mixture-model based applications. The ARI arises from an adjustment made to the Rand index (RI; Rand, 1971) to account for random chance agreement. In other words, the ARI accounts for the fact that random classification would almost certainly classify some observations correctly by chance. The RI is given by the pair agreements,

$$\text{RI} = \frac{\text{number of agreements}}{\text{number of agreements + number of disagreements}},$$

and can take on any value between 0 and 1, with 1 indicating perfect class agreement. The RI however has a positive expected value under random classification, which is what inspired the adjustment to be made. The general form of the ARI is,

$$\text{ARI} = \frac{\text{index} - \text{expected index}}{\text{maximum index} - \text{expected index}}.$$

An ARI value of 1 corresponds to perfect class agreement between two partitions, and the expected value of the ARI under random classification is 0. A negative ARI value indicates classification which is in some sense systematically worse than random classification. There are many other indices other than the ARI available for comparing clustering partitions. The ARI was chosen to assess performance due to extensive presence in past model-based clustering applications.

2.6 *A Posteriori* Merging of Components

In clustering, each cluster is typically represented by one mixture component in a one-to-one relationship, such that one component is analogous to a cluster. However, there are some situations where a cluster may best be represented by a mixture of components rather than a single component. Baudry *et al.* (2010) use mixtures of Gaussian distributions to illustrate certain cases where using a one-to-one mixture component to cluster relationship is not optimal. They argue that a popular selection criterion, the BIC, selects the number of mixture components that best represents the underlying density rather than the true number of components. This can lead to an overestimation of clusters if component densities are taken to represent clusters. For this reason, Baudry *et al.* (2010) propose a hierarchical merging method that first uses the BIC to select a number of mixture components, and then successively merges components based on an entropy criterion. Hennig (2010) also uses mixtures of Gaussian distributions to investigate several hierarchical merging procedures that are based on either modality or misclassification rates. Hennig (2010) illustrates how Gaussian components can be merged for clustering the crabs data set available in the R package (Venables and Ripley, 2002), which contains four true groups. In this example, the BIC originally selects a nine component model, which is optimal for fitting the underlying density, but misleading when each component is interpreted to represent a cluster. See Hennig (2010) for more details on this example as well as other examples of component merging for real and simulated data sets.

It would be expected that there would be more instances where merging is required when clustering with mixtures of densities that are relatively not flexible (ie. Gaussian). In situations where a cluster has a skewed shape, multiple Gaussian components may be needed to represent this cluster, whereas using components such as skew-t or skew-normal distributions have more flexibility to account for this shape and do not require merging. See Vrbik and McNicholas (2014) for illustrations of several data sets available in various R packages where the performance of clustering with a flexible distribution exceeds the performance of merging Gaussian components using the method proposed by Baudry *et al.* (2010).

Wei and McNicholas (2015) use a method for merging components that is based on the maximization of the ARI based on a reference model for mixture model averaging using the GPCM family. In this approach, a reference model is first selected to have the desired number of components, then all combinations of possible component merging are inspected. The merging combination that yields the highest ARI between the reference partition and merging combination is taken to be the best merging combination. Herein, we use the approaches by Wei and McNicholas (2015) for merging components, an example of this merging procedure is provided in Chapter 3 for clarity.

2.7 Bayesian Model Averaging

As mentioned previously, occasionally a model selection criterion such as the BIC may not select a model with the correct number of components or yield the best classification performance, which introduces an element of model selection uncertainty. Additionally, model selection uncertainty is especially amplified for situations where there is not much of a difference between the values of the criteria for two different models. Solely using one model ignores the fact that there may be several models that fit the data very well, but allow for very different inferences. Even though proceeding in this way is the generally accepted norm for model-based clustering applications, recent interest has been placed on combining information from a set of competing models (Wei and McNicholas, 2015), and can be achieved with Bayesian model averaging (BMA; Hoeting *et al.*, 1999). BMA accounts for model selection uncertainty by an approach that utilizes a weighted averaging of models, so that information from several models can be combined. Hoeting *et al.* (1999) use $\mathcal{M}_1$, $\mathcal{M}_2$,...,$\mathcal{M}_K$ to represent the set of all models fitted to the data, and Δ to represent the quantity of interest. The posterior distribution of Δ given data $\mathbf{x}$ is a weighted average of the posterior distribution of Δ under each of the possible models, and can be written,

$$\text{pr}(\Delta \mid \mathbf{x}) = \sum_{i=1}^{K} \text{pr}(\Delta \mid \mathcal{M}_i, \mathbf{x})\text{pr}(\mathcal{M}_i \mid \mathbf{x}) \tag{2.10}$$

where $\text{pr}(\mathcal{M}_i \mid \mathbf{x})$, the posterior probability for model $\mathcal{M}_i$, can be described by,

$$\mathrm{pr}(\mathcal{M}_i \mid \mathbf{x}) = \frac{\mathrm{pr}(\mathbf{x} \mid \mathcal{M}_i)\mathrm{pr}(\mathcal{M}_i)}{\sum_{k=1}^{K} \mathrm{pr}(\mathbf{x} \mid \mathcal{M}_k)\mathrm{pr}(\mathcal{M}_k)} \tag{2.11}$$

and the marginal likelihood of model $\mathcal{M}_i$ is,

$$\mathrm{pr}(\mathbf{x} \mid \mathcal{M}_i) = \int \mathrm{pr}(\mathbf{x} \mid \boldsymbol{\theta}_i, \mathcal{M}_i)\mathrm{pr}(\boldsymbol{\theta}_i \mid \mathcal{M}_i) d\boldsymbol{\theta}_i, \tag{2.12}$$

where $\boldsymbol{\theta}_i$ is the vector of parameters for model $\mathcal{M}_i$, $\mathrm{pr}(\mathcal{M}_i)$ is the prior distribution for $\boldsymbol{\theta}_i$ under model $\mathcal{M}_i$, and $\mathrm{pr}(\mathcal{M}_i)$ is the prior probability of the model $\mathcal{M}_i$. Using all of the models for averaging has been proven to provide a better average predictive ability rather than using a single model (Madigan and Raftery, 1994), however, the implementation of BMA is computationally difficult in a couple of ways:

- The number of models in the summation of (2.10) can be extrordinarily large.
- The posterior model probabilities (2.12) involve very high-dimensional integrals, that are difficult to compute.

In the next two sections, tools for managing these computational issues are provided.

2.8 Occam's Window

To address the issue of too many models in the summation of (2.10), Madigan and Raftery (1994) suggest using Occam's window to select a subset of models for averaging, which provides a significant reduction in the number of candidate models. In this procedure, models that do not lie in the window (ie. models that fit the data far worse than the 'best' model) are discarded and are not subjected to averaging approaches. Occam's window can be written as,

$$\Big\{\mathcal{M}_i : \frac{\max_l\{\text{pr}(\mathcal{M}_l \mid \mathbf{x})\}}{\text{pr}(\mathcal{M}_i|\mathbf{x})} \leq c\Big\}, \tag{2.13}$$

where $\max_l\{\text{pr}(\mathcal{M}_l \mid \mathbf{x})\}$ represents the model with the highest posterior model probability and c is some positive number. The value of c is subjective to the analyst and depends on the context in which BMA is applied. Increasing the value of c increases the size of Occam's window, thereby allowing more models to be eligible for merging. Madigan and Raftery (1994) and Wei and McNicholas (2015) both use $c = 20$ for their analyses, however values of c can generally vary between ten and 100 in the literature. In Chapter 4, values of c between five and 100 will be compared for various real and simulated dat sets to inspect the impact of Occam's window size on clustering performance. Even though increasing the value of c can allow for more models to be eligible for merging, the model weights will reflect when a certain model has just marginally made the 'cutoff'. For instance, models that have a very small difference in selection criterion in comparison to the 'best' model will hold a much larger weight in the averaging process than models that do not fit the data as well but have barely made the window requirements. The next section outlines an approximation for the model weights to resolve computational issues.

2.9 Approximation with the BIC

To address the computational issues of the posterior model probabilities in (2.12), Wei and McNicholas (2015) use an approximation with the BIC (Dasgupta and Raftery, 1998),

$$\text{pr}(\mathbf{x} \mid \mathcal{M}_i) = \exp\Big\{-\frac{1}{2}\text{BIC}_i\Big\}, \tag{2.14}$$

where BIC_i is the BIC value for model $\mathcal{M}_i$, and the model weights can be approximated by:

$$\text{pr}(\mathcal{M}_i \mid \mathbf{x}) \approx \frac{\exp\{-\frac{1}{2}\text{BIC}_i\}}{\sum_{k=1}^{K} \exp\{-\frac{1}{2}\text{BIC}_k\}} \tag{2.15}$$

and Occam's window can be written,

$$\left\{\mathcal{M}_i : \text{BIC}_i - \min_l\{\text{BIC}_l\} \leq 2\text{log}c\right\} \tag{2.16}$$

These approximations, (2.15) and (2.16), will be used for our analyses of real and simulated data sets in Chapter 4.

Chapter 3

Methodology

3.1 Merging Mixture Components

Sometimes the models in Occam's window will have different numbers of components. In these situations, merging mixture components is necessary before averaging is able to take place. To achieve this, we use the method introduced by Wei and McNicholas (2015) which selects a 'reference' model to have the desired number of components, and then merges the other models in Occam's window to match. Models that have less components than the reference model are discarded and only models with the same number of components or more components than the reference model are able to be used for averaging. There are two cases for selecting the reference model: In Case I the model with the smallest BIC ('best' BIC) is the reference model, and in Case II the model with the fewest number of components is the reference model. Under Case I, choosing the reference model to be the model with the best BIC seems to be a natural choice since the BIC has selected that model to be the best fit to the data. Case II accounts for the fact that the BIC may overestimate the number of components, and is meant to try and reduce the number of components of the averaged model if that issue arises. As mentioned previously, the merging criteria for Wei and McNicholas (2015) is based on selecting the merging combination that produces the highest ARI with regards to the reference model. The different merging possibilities were created using the function `combn()` from R package `combinat` (Chasalow, 2012), and the function `permutations()` from R package `permute` (Simpson, 2016).

The density of the merged model is an alternate form of the original model. If the reference model has H components and a model in Occam's window has G ($H < G$) components, then using the same notation as Wei and McNicholas (2015), the density of the new merged model would be,

$$f(\mathbf{x}) = \sum_{j=1}^{H} \pi_j^* f_j^*(\mathbf{x}) = \sum_{g=1}^{G} \phi(\mathbf{x} \mid \boldsymbol{\theta}_g), \tag{3.1}$$

where π_j^* represents one π_j or the sum of several mixing proportions $\pi_1, ..., \pi_G$, and $f_j^*(\mathbf{x})$ represents one $f_j(\mathbf{x})$ or a sum of several component densities from $\phi(\mathbf{x} \mid \ \boldsymbol{\theta}_1), ..., \phi(\mathbf{x} \mid \ \boldsymbol{\theta}_G)$.

3.2 Merging Process Example

The merging process presented by Wei and McNicholas (2015) is best described through an example. In this example, which is analogous to the example given by Wei and McNicholas (2015), suppose the reference model is a three component model and one of the models in Occam's window is a five component model. In other words, the goal is to merge {1, 2, 3, 4, 5} into a model with components {a, b, c}. The partitioning of the reference model is taken to be the 'true' classification and is called the 'reference partition'. The steps are as follows:

- 1. A matrix **A** of size 10×3 is constructed containing the different combinations, $\binom{5}{3}$. In this matrix, each row represents a partial clustering. For instance, consider the 2nd row of A, $\mathbf{a}_2 = (1, 2, 4)$, component 1 would be placed into new component a, component 2 would be placed into new component b, and component 4 would be placed into new component c.

- 2. Next, for each row in **A**, the remaining components that have not yet been assigned to new clusters must be considered. Following our previous example, the components that still need to be assigned to new clusters are $\{3,\ 5\}$. A 9×2 permutation matrix **B** is created to include all the possibilities. Following our same example, row $\mathbf{b}_3 = (a, c)$, places component 3 into component a, and component 5 into component c. The model after merging can now be represented by the new components $\{a, b, c\} = \{1 \cup 3, 2, 4 \cup 5\}$.

- 3. A 10×9 matrix **C** is then used to store the ARI's between the reference partition and all of the partitions arising from the different permutation possibilities of merging. From our example $\mathbf{a}_2 = (1, 2, 4)$ in the second row of matrix **A**, and $\mathbf{b}_3 = (a, c)$ from the 3rd row of matrix **B** would be stored in the second row and 3rd column of matrix **C**.

After all of the ARI's from the different merging partition possibilities have been calculated and stored within matrix **C**, the partition corresponding to the highest ARI value is chosen to be the best merging combination.

3.3 Averaging *A Posteriori* Probabilities

One method used for model averaging involves computing a weighted average of the *a posteriori* probabilities for the models in Occam's window. The process for averaging *a posteriori* probabilities has two steps:

1. Merge models as needed to match the number of components of the reference model.
2. Take the sum of the *a posteriori* probabilities, $\hat{z}_{ig}$, for each model multiplied by their respective weight, and harden to attain predicted classifications.

When two groups are merged together in the first step, the relevant $\hat{z}_{ig}$ values from the corresponding groups are added together as well. Care must be taken to ensure that the groups are aligned and the correct $\hat{z}_{ig}$ are added together. For instance, the resulting *a posteriori* probabilities from merging groups 1, 2, and 3, to produce a new group A would be,

$$\hat{z}_{iA} = \sum_{j=1}^{3} \hat{z}_{ij} \tag{3.2}$$

In the second step, the *a posteriori* probabilities from averaging competing models $\mathcal{M}_1, ..., \mathcal{M}_k$, denoted $\hat{z}^*_{ig}$, can be described by,

$$\hat{z}^*_{ig} = \sum_{m=1}^{k} \text{pr}(\mathcal{M}_m \mid \mathbf{x}) \hat{z}_{ig}^{(\mathcal{M}_m)}, \tag{3.3}$$

where $\mathcal{M}_m$ is used to denote the competing model, $\hat{z}_{ig}^{(\mathcal{M}_m)}$ are the *a posteriori* probabilities (expectations) corresponding to model $\mathcal{M}_m$, $\text{pr}(\mathcal{M}_m \mid \mathbf{x})$ are the weights corresponding to model $\mathcal{M}_m$ (described by (2.15)), and k is the total number of models being averaged. $\hat{z}^*_{ig}$ are then hardened by the process given by (2.9) to produce our predicted classifications.

3.4 Direct Model Averaging

For our second model averaging approach, only the models in Occam's window with the same number of components as the reference model under Case I (ie. the model with the smallest BIC) can be used. This averaging approach involves the direct averaging of model parameters, which is why including merged models would not be possible. A significant difference between the direct model averaging approach and the *a posteriori* model averaging approach is that the former produces a single interpretable model, whereas the latter does not. Consider an example where there are three models in Occam's window; two of them have two components, and the other model has three components. If the model with the smallest BIC is one of the two-component models, then the model with three components must be discarded. Recall the vector of parameters of the generalized hyperbolic distribution under the parametrization of Browne and McNicholas (2015) is: $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\alpha}, \omega, \lambda)$. Let $\boldsymbol{\theta}_1$ represent the vector of parameters from the first model ($\mathcal{M}_1$), and let $\boldsymbol{\theta}_2$ represent the vector of parameters from the second model ($\mathcal{M}_2$). Then, the resulting model parameters from direct model averaging, $\boldsymbol{\theta}^* = (\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*, \boldsymbol{\alpha}^*, \omega^*, \lambda^*)$, are calculated by:

$$\boldsymbol{\theta}^* = \sum_{i=1}^{k} \text{pr}(\mathcal{M}_i \mid \mathbf{x})\boldsymbol{\theta}_i, \tag{3.4}$$

where $k = 2$ in this case, and $\text{pr}(\mathcal{M}_i \mid \mathbf{x})$ are the weights described in (2.15). Then, $\boldsymbol{\theta}^*$ can be used to determine the *a posteriori* probabilities (expected values), $\hat{z}_{ig}$, from (2.8), which are then hardened to attain the predicted classifications.

3.5 Label Switching

For both the *a posteriori* averaging and direct model averaging approaches, care must be taken into aligning the clusters so that components are correctly matched across different models. Stephens (2000) discusses the problem of label switching in mixture-models. For example, one model may label a component as 1, but another model may label that same component as 2. Before averaging is able to take place, a procedure must be used to align these two initially differently labelled clusters so that the same label denotes similar clusters across the base clusterings. In both methods, after models are merged as required, relabelling of components was conducted by a method that created all labelling possibilities of components and selected the relabelled partition that had the greatest similarity to the labelling partition of the reference model.

Chapter 4

Analyses

4.1 Performance and Evaluation

4.1.1 Purpose

In this Chapter, we demonstrate both the averaging *a posteriori* (AAP) and direct model averaging approaches for generalized hyperbolic mixture models. The goal of our model averaging is to improve upon the clustering performance of the single 'best' model, as determined by the BIC. In addition to inspecting the performance of model-averaging versus the performance of the 'best' model, it was also of interest to explore how the size of Occam's window impacted the clustering performance. As the value of c in (2.13) is increased, Occam's window expands allowing more models to be eligible for merging. The farther a model fit is from the fit of the best model is reflected in a much smaller associated model weight. Values of c that were tested included 5, 20, 40, 60, 80, and 100.

4.1.2 Generation of Clustering Models

To generate a group of models for averaging, the function `MGHD()` from the `R` package `MixGHD` (Tortora *et al.*, 2015) was used to carry out model based clustering using mixtures of generalized hyperbolic distributions. Different settings of number of groups to fit to the data and different initialization strategies for the algorithm were inspected creating a variety of models. The different initialization strategies implemented include:

- ***k*-means Clustering:** Partitions observation points such that the sum of squares from points to the assigned clusters is minimized (Steinley, 2006).
- **Hierarchical:** First, each observation is assigned to its own cluster, then the two closest clusters are joined to form a single cluster. The process is repeated until there is only one cluster.
- **Random:** Randomly assigns each observation point to a group.
- **Model Based clustering:** The data are clustered using a mixture modelling structure.

4.2 Simulation Scenarios

Two simulated data scenarios are considered in this section. The function `genRandomClust()` from the `R` package `clusterGeneration` (Qiu and Joe., 2015) was used to generate random clusters in both scenarios.

4.2.1 Scenario I: Easy Clustering Problem

In Scenario I, a data set was generated with $p = 3$ variables, $n = 480$ observations, and $G = 3$. Setting `sepVal=0.2`, `numNonNoisy=3`, and `clustzind=2` created a easy clustering problem with three non noisy variables and clusters spread fairly apart (Figure 4.1).

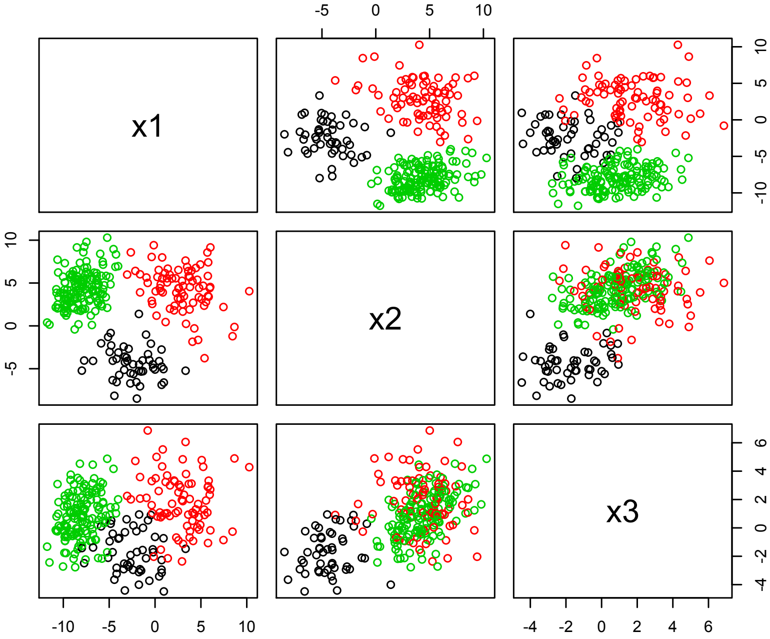

Figure 4.1: Pairs plot for the simulated data in Scenario I.

Models were fit using `MixGHD` for $G = 2, ..., 5$ paired with each of the different initialization strategies for a total of 16 models. Changing the value of c did not impact the number of models in Occam's window, and each window size contained the same three models: k-means initialization with $G = 3$, hierarchical initialization with $G = 3$, and model-based clustering initialization with $G = 3$. The BIC selected the model-based clustering initialization with $G = 3$ as the 'best' model, and merging was not required since all of the models in Occam's window had the same number of components (Table 4.1).

Table 4.1: A summary of the models in Occam's window along with ARI values for the true labels versus predicted classifications from the best model, from AAP, and from direct model averaging (MA), respectively, for the simulated data in Scenario I (for $c = 5, 20, 40, 60, 80, 100$).

Occam's window			$\Pr(\mathcal{M}_i \mid D)$	ARI values			
Model Init.	BIC	G		Best	AAP		MA
					Case I	Case II	
k-means	2007.38	3	0.3342	0.9750	0.9874	0.9874	0.8335
Hier.	2007.61	3	0.2982				
Mod	2007.19	3	0.3676				

Because $G = 3$ for all models in Occam's window, AAP under Case 1 and Case 2 were identical. AAP achieved marginal improvement in classification performance (ARI = 0.9874), whereas there was decrease in performance for the direct model averaging method (ARI = 0.8335). All methods performed well as was expected for the level of difficulty of this scenario.

4.2.2 Scenario II: Hard Clustering Problem

Scenario II generated a data set with $p = 3$ variables, $n = 480$ observations with $G = 4$. Setting `sepVal=0.03`, `numNonNoisy=3`, and `clustzind=2`. These settings created a difficult clustering problem with clusters overlapping significantly. Scenario II was generated to illustrate our merging approach's ability to enhance the classification performance when observation points lie at the intersection of clusters, and may otherwise be misclassified. Figure 4.2 illustrates this overlap with colour denoting true group membership. Models were fit for $G = 2, ..., 5$ paired with each of the different initialization strategies discussed previously for a total of 16 models. In this scenario, the the number of models in Occam's window increased as the value of c increased as shown in Table 4.2.

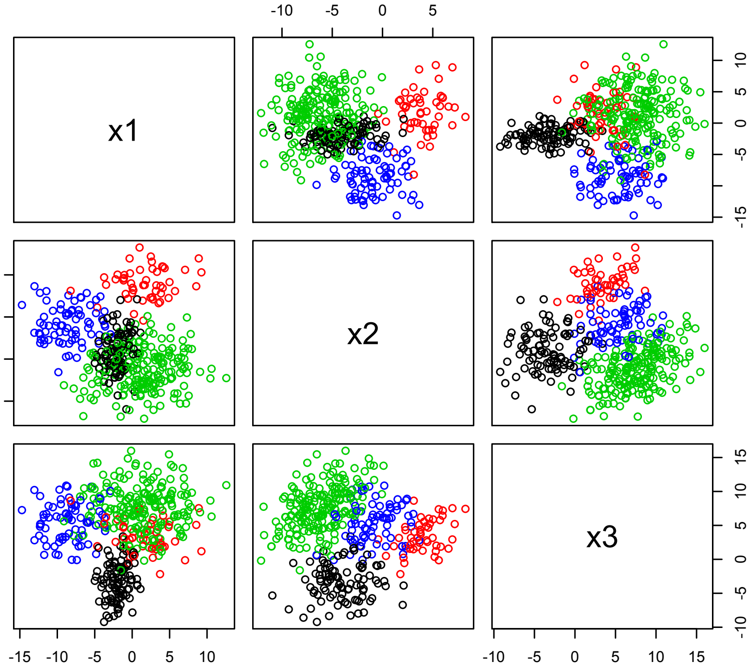

Figure 4.2: Pairs plot for the simulated data in Scenario II.

For c set to 100, 80, 60, and 40, Occam's window selected five of the models: k-means initialization with $G = 3$, k-means initialization with $G = 4$, hierarchical initialization with $G = 3$, hierarchical initialization with $G = 4$, and model-based initialization with $G = 3$. The three component hierarchical initialization model was selected to be the 'best' model in terms of the BIC (3282.09). Because the 'best' model is also the model with the least number of components, the results from Case I and Case II AAP are identical. Both averaging *a posteriori* probabilities (ARI = 0.8327) and model averaging (ARI = 0.8488) led to a slight improvement in classification performance, with the latter being slightly better (Table 4.3).

Table 4.2: Models in Occam's Window for Different Window Sizes for Simulated Data Scenario II.

Value of c	Number of Models in Occam's Window
100	5
80	5
60	5
40	5
20	4
5	3

Table 4.3: A summary of the models in Occam's window along with ARI values for the true labels versus predicted classifications from the best model, from AAP, and from direct model averaging (MA), respectively, for Scenario II (for $c = 40, 60, 80, 100$).

Occam's window			$\Pr(\mathcal{M}_i \mid D)$		ARI values			
Model Init.	BIC	G	Case I/II	MA	Best	AAP		MA
						Case I	Case II	
k-means	3283.27	3	0.2606	0.2730	0.8245	0.8327	0.8327	0.8488
k-means	3288.21	4	0.0219					
Hier.	3282.09	3	0.4682	0.4906				
Hier.	3288.06	4	0.0237					
Mod.	3283.55	3	0.2255	0.2363				

Four models lie in Occam's window when c is set to 20: k-means initialization with $G = 3$, hierarchical initialization with $G = 3$, hierarchical initialization with $G = 4$ and model-based initialization with $G = 3$. The k-means initialization with $G = 4$ model is the only model excluded in comparison to the previous window settings. However, since it's weight was so small, the results are not impacted greatly by its exclusion (Table 4.4).

Reducing the window size further by setting $c = 5$ reduces the number of models in Occam's window to three: k-means initialization with $G = 3$, hierarchical initialization with $G = 3$, and model-based initialization with $G = 3$. Because $G = 3$ for all models, AAP under Case I and Case II are identical. Classification performance

Table 4.4: A summary of the models in Occam's window along with ARI values for the true labels versus predicted classifications from the best model, from AAP, and from direct model averaging (MA), respectively, for Scenario II (for $c = 20$).

Occam's window			Pr($\mathcal{M}_i \mid D$)		ARI values			
Model Init.	BIC	G	Case I/II	MA	Best	AAP		MA
						Case I	Case II	
k-means	3283.27	3	0.2664	0.2730	0.8245	0.8327	0.8347	0.8488
Hier.	3282.09	3	0.4787	0.4906				
Hier.	3288.06	4	0.0242					
Mod.	3283.55	3	0.2306	0.2363				

was not impacted greatly by changing the size of the window, and classification performance was marginally improved for all merging methods (Table 4.5). It is noticed that even though the true number of groups in the data is four, the window setting with $c = 5$ only selects models with $G = 3$.

Table 4.5: A summary of the models in Occam's window along with ARI values for the true labels versus predicted classifications from the best model, from AAP, and from direct model averaging (MA), respectively, for Scenario II (for $c = 5$).

Occam's window			Pr($\mathcal{M}_i \mid D$)	ARI values			
Model Init.	BIC	G		Best	AAP		MA
					Case I	Case II	
k-means	3283.27	3	0.2730	0.8245	0.8327	0.8347	0.8488
Hier.	3282.09	3	0.4906				
Mod.	3283.55	3	0.2363				

A summary of the all the classification performances for scenario II is provided in Table 4.6. In all sizes of Occam's window, marginal improvement is seen from all

Table 4.6: Summary of classification results for Simulated Scenario II.

Value of c	AAP Case I	AAP Case II	MA
40, 60, 80, 100	0.8327	0.8327	0.8488
20	0.8327	0.8347	0.8488
5	0.8327	0.8347	0.8488

averaging approaches in comparison to the performance of the 'best' model. Because the 'best' model was a three component model as well as the model with the least number of components in Occam's window, every averaging approach underestimated the number of groups in the data. The benefit of viewing models in Occam's window is that it brings awareness to other models that could possibly have the correct number of components. A larger window size (where c= 40, 60, 80, and 100) included four-component models whereas smaller window sizes did not. However the corresponding weights for these four-component models were extremely small.

4.3 Applications to Datasets

4.3.1 Yeast

The yeast data set concerns yeast cellular localization sites of 1,484 proteins and can be found in the UCI machine learning repository (Nakai and Kanehisa, 1991). Three variables were considered for this analysis: McGeoch's method for signal sequence recognition (MCG), the score of the ALOM membrane spanning region prediction program (ALM), and the score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins (VAC). There are two true localization sites in the data set ($G = 2$): CYT (cytosolic or cytoskeletal) and ME3 (membrane protein, no N-terminal signal) (Figure 4.3).

Models were fit using `MixGHD` for $G = 2, ..., 5$ paired with each of the different initialization strategies for a total of 16 models. Occam's window selects two models for larger sizes of Occam's window and one model for a smaller sizes of Occam's window (Table 4.7).

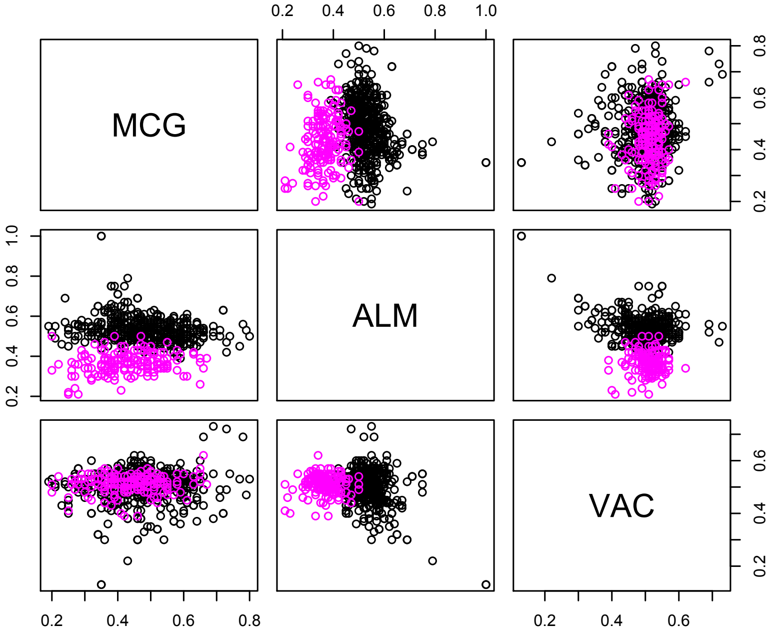

Figure 4.3: Pairs plot for the yeast data.

For c = 40, 80, and 60, two models lie in Occam's window: hierarchical initialization with G = 2, and model-based initialization with G = 3. Model-based initialization with $G = 3$ was selected as the 'best' model by the BIC (5036.84). For AAP, we must either ignore model hierarchical initialization with G = 2 (Case I) or merge the components of the model-based initialization model to produce a two-component model so that both models can be used (Case II). AAP under Case II leads to a large improvement in classification performance (ARI of 0.8257). There is only one model eligible under Case I, so averaging does not take place and the classification performance is identical to that of the 'best' model (ARI of 0.5531). When directly averaging models, again only one model is eligible leading to a classification performance that is identical to that of the 'best' model (ARI of 0.5531) (Table 4.8).

Table 4.7: Models in Occam's Window for Different Window Sizes for the Yeast Data.

Value of c	Number of Models in Occam's Window
100	2
80	2
60	2
40	1
20	1
5	1

Table 4.8: A summary of the models in Occam's window along with ARI values for the true labels versus predicted classifications from the best model, from AAP, and from direct model averaging (MA), respectively, for the yeast data (for $c = 60, 80, 100$).

Occam's window			$\Pr(\mathcal{M}_i \mid D)$	ARI values			
Model Init.	BIC	G	Case II	Best	AAP		MA
					Case I	Case II	
Hier.	5044.31	2	0.0233	0.5531	0.5531	0.8134	0.5531
Mod.	5036.84	3	0.9767				

For $c = 5$, 20, and 40 only one model, model-based initialization with $G = 3$, lies in Occam's window. Therefore, no averaging is able to take place and the classification performance is identical to that of the 'best' model (ARI = 0.5531). Table 4.9 provides a summary of the classification performance for all of the averaging approaches versus the best model. AAP with a larger window size (c = 60, 80, and 100) under Case II achieved a greatly superior classification performance in comparison to the 'best' model (ARI = 0.8134). The BIC overfit the data, which lead to an overestimation of the number of components in the 'best' model. When averaging *a posteriori* under Case II the model in Occam's window with the least number of components, hierarchical initialization with $G = 2$, was the reference model. The other model in Occam's window was merged to match for $G = 2$, and because there are two true groups in the yeast data, the classification performance was greatly improved.

Table 4.9: Summary of results for different values of c for the yeast data set.

Value of c	AAP Case I	AAP Case II	MA
60, 80, 100	0.5531	0.8134	0.5531
5, 20, 40	0.5531	0.5531	0.5531

4.3.2 AIS

The Australian Institute of Sport (AIS) data concerns eleven biometric measurements for athletes and is available in the `alr3` package for `R` (Weisberg, 2010). There are two true groups in this data set, with 102 male and 100 female athletes. The eleven biometric measurements can be found in Table 4.10. A pairs plot of the data indicates a slight overlapping of the two groups (Figure 4.4).

Table 4.10: Biometric Variables of the AIS data set used for analysis.

Height
Weight (kg)
Lean body mass
Red cell count
White cell count
Hematocrit
Hemoglobin
Plasma ferritin concentration
Body mass index
Sum of skin folds
Percent body fat

Mixtures of generalized hyperbolic distributions were fitted to these data using `MixGHD` for $G = 2, ..., 5$, and the four different initialization strategies, for a total of sixteen different models. The size of Occam's window did not impact the number of models that were selected for averaging. For all values of c, Occam's window selected four models: k-means initialization with $G = 2$, k-means initialization with $G = 3$, hierarchical initialization with $G = 2$, and model based initialization with $G = 2$. The BIC selected k-means initialization with $G = 3$ to be the 'best' model (2282.07). For AAP, either all of the two-component models must be ignored (Case I) or the 'best' model must be merged into a two-component model in order to use all of the models in Occam's window for averaging (Case II). For the direct model averaging approach, the

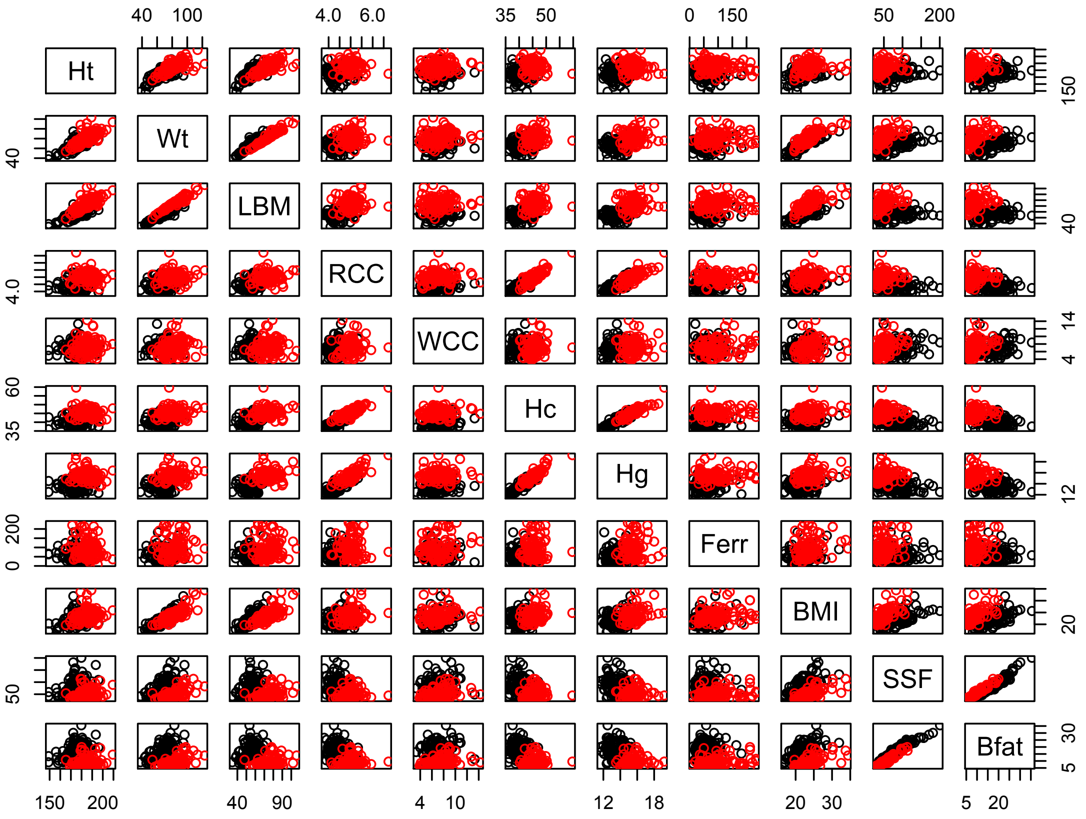

Figure 4.4: Pairs plot for the AIS data.

'best' model has more components than all other models in Occam's window; therefore the other models are discarded, and averaging does not take place. The results from all averaging approaches in comparison to the classification performance of the 'best' model is provided in Table 4.11. AAP under Case I and direct model averaging lead to the same classification performance as given by the 'best' model since the 'best' model was the only model eligible for averaging (ARI = 0.6034). AAP under Case II produces a great improvement in classification performance was observed (ARI = 0.9030). Here, the reference model had the same number of components as the true number of groups in the data set. The other models in Occam's window were merged to match before averaging took place, which resulted in a greatly increased classification performance.

Table 4.11: A summary of the models in Occam's window along with ARI values for the true labels versus predicted classifications from the best model, from AAP, and from direct model averaging (MA), respectively, for the AIS data (for $c = 40, 60, 80, 100$).

Occam's window			$\Pr(\mathcal{M}_i \mid D)$	ARI values			
Model Init.	BIC	G	Case II	Best	AAP		MA
					Case I	Case II	
k-means	2282.93	2	0.2177	0.6034	0.6034	0.9030	0.6034
k-means	2282.07	3	0.3353				
Hier	2282.85	2	0.2271				
Mod.- based	2282.90	2	0.2199				

4.3.3 Examples with only one model in Occam's window: crabs data, and wine data

The crabs data set contains five morphometric measurements on two species of Leptograpsus crabs (blue and orange), that are also separated into two genders. The morphometric measurements include: frontal lobe size, carapace length, body depth, rear width, and carapace width. These data are available in the `MASS` library for `R` (Venables and Ripley, 2002).

The wine data set can be found in `R` package `gclus` (Hurley, 2012). This data set consists of 13 physical and chemical measurements on 178 Italian wines from the same region, but from three different cultivars: Barolo, Grignolino, and Barbera ($G = 3$).

Mixtures of generalized hyperbolic distributions were fit to each data set via `MGHD()` for $G = 2, .., 5$, and each analysis resulted in only one model lying within

Occam's window. For the crabs data set, the BIC chose the model with random initialization and $G = 3$ components. For the wine data set, the model with hierarchical initialization and $G = 2$ was selected as the best model by the BIC. The 'best' models selected by the BIC for both data sets had less components than the true number of groups. Naturally, when there is only one model in Occam's window, averaging is equivalent to reporting classifications from the model with the best BIC.

Chapter 5

Conclusions

This thesis extended the mixture model averaging methods by Wei and McNicholas (2015) to averaging with non-Gaussian, mixtures of generalized hyperbolic distributions. Averaging *a posteriori* and direct averaging of model parameters were the two model averaging methods that we explored. To resolve computational issues, two tools were leveraged: Occam's window, and an approximation with the BIC. Occam's window selected a subset from the total number of models that had similar performance to the model with the smallest BIC, and greatly decreased the number of models to average. Additionally, an approximation with the BIC eased the computational difficulty of the model weights. When models in Occam's window had different numbers of components, a method based on the ARI was used to merge components in regards to a reference model. Care was taken to ensure that the correct components were aligned across the competing models before merging components and averaging models.

Wei and McNicholas (2015) achieved positive results from averaging approaches, and here we saw enhanced classification performance from these averaging approaches as well. Two simulation scenarios and four real data sets were used to illustrate the performance of the different averaging techniques. In comparison to the analyses of Wei and McNicholas (2015), it was expected that we would have to merge components less than if the components were Gaussian. A flexible density such as the generalized hyperbolic can potentially model a skewed cluster that would require several Gaussian components to model. For both of the simulated data sets, the performance of averaging *a posteriori* under both cases was slightly better than that of the best model. The direct model averaging approach also performed better for the second simulated scenario. For the yeast data set and the AIS data set, AAP under Case II performed the best because the BIC overestimated the number of components for both data sets. In the crabs and wine data sets, only one model was selected by Occam's window, and averaging was equivalent to reporting classifications from the model with the best BIC. In many situations, it was also seen that larger values of c for Occam's window were the most beneficial for increasing classification performance. AAP under Case II tended to have the best overall performance in comparison to the other averaging methods. AAP under Case II does not discard any models in Occam's window whereas the other methods do, which may be the driving force behind this increased performance. However, AAP under Case II can potentially underestimate the number of components.

Future work can consist of model averaging with other distributions, such as mixtures of skew-t distributions. Additionally, future work can investigate mixture model averaging in a semi-supervised classification framework.

Bibliography

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

Barndorff-Nielsen, O. and Halgreen, C. (1977). Infinite divisibility of the hyperbolic and generalized inverse Gaussian distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, **38**, 309–311.

Barndorff-Nielsen, O. E. (1997). Normal inverse Gaussian distributions and stochastic volatility modelling. *Scandinavian Journal of Statistics*, **24**(1), 1–13.

Baudry, J.-P., Raftery, A. E., Celeux, G., Lo, K., and Gottardo, R. (2010). Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, **19**(2), 332–353.

Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(7), 719–725.

Blæsild, P. (1978). The shape of the generalized inverse Gaussian and hyperbolic distributions. Research Report 37, Department of Theoretical Statistics, Aarhus University, Denmark.

Browne, R. P. and McNicholas, P. D. (2015). A mixture of generalized hyperbolic distributions. *Canadian Journal of Statistics*, **43**(2), 176–198.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, **28**(5), 781–793.

Chasalow, S. (2012). *combinat: combinatorics utilities*. R package version 0.0-8.

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

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

Franczak, B., Browne, R., McNicholas, P., and Findlay, C. (2015). Product selection for liking studies: The sensory informed design. *Food Quality and Preference*.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, **40**, 237–260.

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

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, **14**(4), 382–401.

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

Hurley, C. (2012). *gclus: Clustering Graphics.* R package version 1.3.1.

Jørgensen, B. (1982). *Statistical Properties of the Generalized Inverse Gaussian Distribution.* Springer-Verlag, New York.

Madigan, D. and Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, **89**(428), 1535–1546.

McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques and Tools.* Princeton University Press.

McNicholas, P. D. (2016a). *Mixture Model-Based Classification.* Chapman & Hall/CRC Press, Boca Raton.

McNicholas, P. D. (2016b). Model-based clustering. *Journal of Classification*, **33**(3), 331–373.

McNicholas, P. D. and Subedi, S. (2012). Clustering gene expression time course data using mixtures of multivariate t-distributions. *Journal of Statistical Planning and Inference*, **142**(5), 1114–1127.

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

Nakai, K. and Kanehisa, M. (1991). Expert system for predicting protein localization sites in gram-negative bacteria. *Proteins*, **11**(2), 95–110.

Qiu, W. and Joe., H. (2015). *clusterGeneration: Random Cluster Generation (with Specified Degree of Separation)*. R package version 1.3.4.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, **66**(336), 846–850.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann Stat*, **6**, 461–464.

Simpson, G. L. (2016). *permute: Functions for Generating Restricted Permutations of Data*. R package version 0.9-4.

Steinley, D. (2006). K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, **59**, 1–34.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B*, **62**(4), 795–809.

Tortora, C., Browne, R. P., Franczak, B. C., and McNicholas, P. D. (2015). *MixGHD: Model Based Clustering, Classification and Discriminant Analysis Using the Mixture of Generalized Hyperbolic Distributions*. R package version 1.8.

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

Vrbik, I. and McNicholas, P. D. (2014). Parsimonious skew mixture models for model-based clustering and classification. *Computational Statistics and Data Analysis*, **71**, 196–210.

Wei, Y. and McNicholas, P. D. (2015). Mixture model averaging for clustering. *Advances in Data Analysis and Classification*, **9**(2), 197–217.

Weisberg, S. (2010). *alr3: Companion to Applied Linear Regression*. R package version 2.0.

Wolfe, J. H. (1965). A computer program for the maximum likelihood analysis of types. Technical Bulletin 65-15, U.S. Naval Personnel Research Activity.