On Supervised and Unsupervised Discrimination

by

Will Ruth

B.Sc., Simon Fraser University, 2014

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2016

Approval

Name:	**Will Ruth**
Degree:	**Master of Science (Statistics)**
Title:	***On Supervised and Unsupervised Discrimination***
Examining Committee:	**Chair:** Dr. Tim Swartz Professor

Dr. Tom Loughin
Senior Supervisor
Professor

Dr. Richard Lockhart
Supervisor
Professor

Dr. Liangliang Wang
Internal Examiner
Assistant Professor

Date Defended: July 29, 2016

Abstract

Discrimination is a supervised problem in statistics and machine learning that begins with data from a finite number of groups. The goal is to partition the data-space into some number of regions, and assign a group to each region so that observations there are most likely to belong to the assigned group. The most popular tool for discrimination is called discriminant analysis. Unsupervised discrimination, commonly known as clustering, also begins with data from groups, but now we do not necessarily know how many groups, nor do we get to know which group each observation belongs to. Our goal when doing clustering is still to partition the data-space into regions and assign groups to those regions, however we do not have any a priori information with which to assign these groups. Common tools for clustering include the k-means algorithm and model-based clustering using either the expectation maximization (EM) or classification expectation maximization (CEM) algorithms (of which k-means is a special case).

Tools designed for clustering can also be used to do discrimination. We investigate this possibility, along with a method proposed by Yang (2013) for smoothing the transition between these problems. We use two simulations to investigate the performance of discriminant analysis and both versions of model-based clustering with various parameter settings across various datasets. These settings include using Yang's method for modifying clustering tools to handle discrimination. Results are presented along with recommendations for data analysis when doing discrimination or clustering. Specifically, we investigate what assumptions to make about the groups' sizes and shapes, as well as which method to use (discriminant analysis or the EM or CEM algorithms) and whether or not to apply Yang's pre-processing procedure.

Keywords: Clustering; Classification; Supervised learning

Acknowledgements

I would first like to thank my supervisor, Dr. Tom Loughin, for his continual support and patience throughout my degree. I am particularly grateful for all of his feedback on this document. More generally, Tom has been an excellent mentor, and has undoubtedly played a crucial role in making me the researcher I am today.

I would also like to thank all the other members of the department, from faculty to staff to other students. Our department is a wonderful place to live and grow, and my life is richer for everyone in it. I look forward to spending more time here in the future, and getting to continue working with these excellent people.

No acknowledgments section would be complete without thanking my awesome fiancé, Jaclyn, and the rest of my wonderful family and friends for their continual love and support through the sometimes trying experience of my masters program.

Finally, I would like to thank SFU, the Statistics and Actuarial Science department, and the National Science and Engineering Research Council for their financial support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduce Problems

1.1.1 What is Clustering?

Clustering is a well-known unsupervised learning problem where we have n observations in $\mathbb{R}^d$, and we want to assign them to groups, or "clusters", based on which observations are similar. We assume that each observation comes from one of a small number of groups, where the population within each group is different in some way (Hastie et al., 2011). Typically, the groups have different means, but their variances or some other features may differ as well. We must then create clusters based on our data with the hope that these clusters accurately reflect the population groups. Clusters are usually chosen in practice so that an observation is more similar to its own cluster than to other clusters. A challenging aspect of this problem however, is that we have no a priori information about the groups, so we must simultaneously construct clusters and assign observations to them. For example, suppose that we have n cancer patients, each with values on d different clinical measurements. We may then want to cluster patients together so that we can tell whether a particular person has a common form of cancer or a rare one, according to the relative sizes of the clusters. Different clustering algorithms are chosen both for how they measure similarity, and how they use this measure to construct clusters. An assignment of observations to clusters is called a "solution" to the clustering problem, or a "clustering solution".

1.1.2 What is Discrimination?

Discrimination or discriminant analysis (DA) is a similar problem, where we have the same n observations in $\mathbb{R}^d$ as for clustering, but we now have labels that tell us which group each observation comes from. In the cancer example, suppose that we want to determine the characteristics of patients who experience different treatment outcomes. Here we not only have information about the patients, but we also know what outcome they experience. The goal is then to partition d-dimensional space into disjoint regions such that points in the same region tend to come from the same group (Hand, 1981). These regions are similar to the clusters constructed by clustering algorithms. Note that DA is related to another problem called classification, where the same data structure is used to predict the group label of a new observation (Hand, 1981).

By far the most common methods for doing discrimination assume that data are normally distributed with some or all of the parameters allowed to differ across groups (Hastie et al., 2011). After estimating the parameters in each group, discrimination is done by assigning each point in $\mathbb{R}^d$ to the group for which it has the highest probability of membership. Recall that the multivariate normal distribution is parameterized by a mean and a covariance matrix (Wasserman, 2004). If we make no assumptions about the groups' covariance matrices, the resulting process is called quadratic discriminant analysis, or QDA, since its decision boundaries (i.e., the boundaries between any two adjacent groups) are quadratic curves. Another, simpler version of DA is called linear discriminant analysis, or LDA. This method assumes that the groups have identical covariance matrices, and is therefore a more parsimonious model than QDA. The decision boundaries in LDA are, perhaps unsurprisingly, linear.

1.2 Motivation and Outline

It is apparent that clustering and discrimination are very similar problems. They both work with similar data structures, and solutions to both problems are of the same form (a partition of d-space into disjoint clusters or regions). The only difference is whether or not we know which groups the observed individuals belong to. For the rest of this thesis, we use "groups" to refer to the population categories, and "clusters" to refer to their sample analogues.

It is natural to wonder if an algorithm developed for clustering might be able to overcome its lack of a priori information about groups and adequately solve the discrimination problem. The supervised clustering algorithm proposed by Yang (2013) is of particular interest, as it is a supervised learning method that involves using information about the group labels to modify a common clustering algorithm to solve discrimination problems. Some work has been done by O'Neill (1978) to address the relationship between clustering and discrimination, but his paper deals with abstract and asymptotic results. We address more concrete problems related to performance metrics that are empirically measurable.

The goal of this thesis is to investigate in detail the possibility of using clustering to do discrimination. In Chapter 2 we introduce several clustering methods, including the one by Yang, and discuss both their algorithmic and statistical properties. We also introduce standard methods for doing discrimination in this chapter so that we have some benchmark procedures for evaluating clustering algorithms. In Chapter 3, we present two simulation studies comparing the performance of several clustering and discrimination methods, with respect to their ability to accurately partition $\mathbb{R}^d$ into regions where each population group is predominant.. Finally, in Chapter 4, we discuss the results of these simulations, their implications for using clustering to solve discrimination problems and some future work to be done in this area.

Chapter 2

Details and Properties of Methods

2.1 Clustering

Recall that the problem of clustering is to take a sample of n observations in $\mathbb{R}^d$, and use them to partition d-space into disjoint clusters. All clustering methods are based on putting "similar" points in the same cluster. To do this, most methods consist of a metric for similarity and an algorithm for partitioning the sample space based on this metric. It is often convenient to measure the "dissimilarity" between two observations, and we sometimes construct a dissimilarity matrix (Hastie et al., 2011) that functions as a lookup table for the dissimilarity between two observations. That is, element i, j is the dissimilarity between the ith and jth observations. Note that such a matrix is necessarily symmetric. A more common approach, and the one used by the methods we consider, is to define a function for computing the dissimilarity between any two points. For example, the popular k-means algorithm measures the dissimilarity between any two points in $\mathbb{R}^d$ by the d-dimensional Euclidean distance between them (MacQueen, 1967). In general however, the partitioning algorithm can be applied with other dissimilarity metrics (Loohach and Garg, 2012; Melnykov and Melnykov, 2014). If desired, the Euclidean distances between all pairs of observations can be calculated and used to populate a dissimilarity matrix, but this computation is rarely carried out in practice.

There are many algorithms that solve the clustering problem, but they can be broadly classified into two categories, hierarchical and non-hierarchical (Hastie et al., 2011). When constructing multiple clustering solutions with different numbers of clusters (for example, to choose an appropriate number of clusters), hierarchical algorithms require that a solution with k clusters be obtained either by splitting one of the clusters in the $k-1$ cluster solution or by combining two clusters in a k+1 cluster solution. Alternatively, non-hierarchical clustering algorithms have no such requirements and construct new clustering solutions with no reliance on previous ones.

The most popular clustering algorithm is a non-hierarchical one called k-means. We discuss this algorithm in detail in the next section, and all of the more complicated clustering methods we discuss are generalizations of k-means.

2.1.1 Methods

k-Means

The k-means algorithm is very popular for solving clustering problems due to its intuitive appeal and ease of computation (Jain, 2010). Given the number of clusters to be found, k, the algorithm alternates between finding the center of each cluster, called a centroid, and assigning each observation to the cluster with the nearest centroid. The centroid of each cluster is computed as the arithmetic mean of the points in that cluster. The algorithm is initialized either by forming initial clusters or by choosing k locations in d-space as the initial centroids. Both forms of initialization are usually performed randomly. The algorithm then iterates between assignment and computation steps until a suitable convergence criterion is met. Convergence of this type is guaranteed (Celeux and Govaert, 1992), but need not be to a globally optimal solution. The solution obtained is known to be sensitive to initial conditions, so it is common to re-run the algorithm some number of times with random assignments of points to clusters or random initial centroids, and select the solution with smallest within-cluster dissimilarity, an analogue of the sum of squares for error in regression (in k-means this is the sum of squared distances between each point and its respective centroid).

Numerous small variations of this algorithm give different clustering solutions. For example, the "soft k-means" algorithm arises from allowing points to have soft (i.e., proportional) membership in each cluster rather than hard (0-1) membership. The soft k-means algorithm turns out to be related to the EM algorithm, and setting up the framework to make this connection allows us to formulate many more generalizations of the k-means algorithm (Celeux and Govaert, 1995). We discuss the relationship between k-means and the EM algorithm in more detail later.

It is common to standardize each variable to have mean 0 and standard deviation of 1 before running the k-means algorithm. This ensures that each variable contributes equally to the Euclidean distance metric. If this standardization is not performed, then a variable measured in nanograms (10^{-9} grams) might dominate the dissimilarity measure, and therefore the clustering; particularly when the variable ought to be measured in kilograms (1 kilogram = 10^{12} nanograms). Note that standardizing the data implicitly makes the assumption that all variables are equally important.

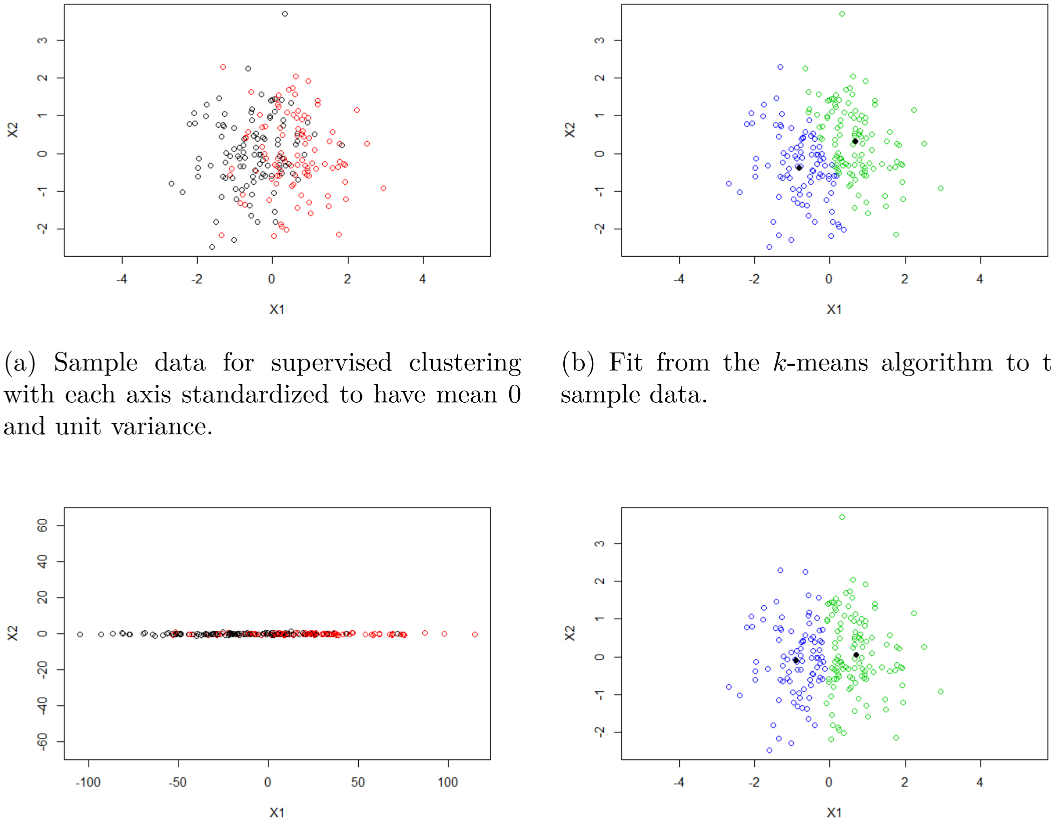
Yang's Supervised Clustering

In a 2013 masters thesis, Yang (2013) proposes a modification of the k-means algorithm to make it perform better when solving discrimination problems. This modification is quite straightforward, and has strong intuitive appeal but he presents little in the way of theoretical justification. To motivate the approach he proposes, consider the bivariate dataset with two groups shown in Figure 2.1a, where observations from group 1 are colored black and observations from group 2 are colored red. These data were generated from two bivariate normal populations with equal covariance matrices and whose means differ only in the X_1 direction. A k-means clustering fit to these data, which ignores information about the groups, gives the cluster assignments and estimated centroids in Figure 2.1b. This is not a particularly good fit to the two groups, since the irrelevant X_2 component has a strong influence on the clustering solution. Yang proposes that before running k-means, we re-scale each predictor variable by an amount proportional to some measure of its univariate ability to discriminate between the groups. Applying this here gives the data shown in Figure 2.1c. Fitting a k-means clustering on this transformed data and back-transforming gives the much better discriminator and cluster centroids in Figure 2.1d.

The way to measure a predictor's ability to discriminate between groups, and the way to use this to scale the corresponding axis, are left somewhat open by Yang. While he does give some recommendations based on desirable properties of the chosen method, there is no discussion of what is "optimal", or even how to characterize optimality. Yang's approach uses a univariate logistic regression fit of Y on each predictor variable individually. We then get out the p-value for the slope coefficient and take its negative logarithm. This is used as a scaling constant for the variable under which it was generated. The negative logarithm transformation is used here because it is a decreasing function of the p-value (therefore an increasing transformation of the level of significance), it compresses the full domain of traditionally "non-significant" p-values into a fairly narrow range, and its growth is moderate as p-values approach 0.

Yang (2013) gives results from a limited number of simulations and data analyses. Specifically, he focuses on predictors from a real-world dataset with actual or simulated class labels, and compares his method primarily to classification trees, random forests (Hastie et al., 2011) and another supervised clustering algorithm called Single Representative Insertion/Deletion Steepest Descent Hill Climbing with Randomized Restart (Eick et al., 2004). Yang's simulations suggest that his method performs well compared to these models. We investigate his procedure relative to other clustering methods, and on a different class of datasets.

There is a known problem with the fitting procedure for logistic regression models, called complete separation. This occurs when the two groups can be perfectly distinguished using the predictor variables. Although this is actually a desirable phenomenon, it can lead



(a) Sample data for supervised clustering with each axis standardized to have mean 0 and unit variance.

(b) Fit from the *k*-means algorithm to the sample data.

(c) Sample data after axis scaling.

(d) Fit from the *k*-means algorithm to the scaled sample data and back-transformed to the original sample data.

Figure 2.1: Data and *k*-means fits before and after axis scaling.

to problems with fitting logistic regression models. Specifically, the maximum likelihood estimates of our slope coefficients may be unbounded (Bilder and Loughin, 2015). Further, the standard errors of these slope parameters can also be unbounded, leading to meaningless hypothesis tests. Heinze and Schemper (2002) propose several ways to address this problem, the most effective of which is based on the penalized likelihood method described in Firth (1993). This Firth likelihood (FL) method involves introducing some bias to the maximum likelihood procedure, but is shown by Heinze and Schemper (2002) to always yield finite parameter values. The standard errors obtained by FL are also much more sensible, so we can obtain reasonable p-values from Wald tests (Wasserman, 2004). We therefore use the p-value from a Wald test of the estimate obtained using the FL fitting procedure in place of the p-value obtained from a simple logistic regression when using Yang's supervised clustering method.

The EM and CEM Algorithms

Recall that the clustering problem consists of assigning observations to clusters without any a priori knowledge about the underlying groups. If we assume that the group labels are fixed but unknown then this can be seen as an incomplete, or missing data problem. Specifically, the unobserved group labels are the missing data.

The EM algorithm (Dempster et al., 1977; Wu, 1983) is a well known method for solving incomplete data problems. While its general properties are well understood (McLachlan and Krishnan, 2008), we are particularly interested here in its application to studying mixture distributions (McLachlan and Peel, 2000). In particular, the soft k-means algorithm discussed above can be seen as a particular version of the EM algorithm applied to solving this mixture problem. Before discussing this connection, we introduce how the EM algorithm operates in the context of mixture distributions.

To begin, assume that we have data from a finite mixture of G groups. That is,

$$F(x) = \sum_{g=1}^{G} p_g F_g(x), \tag{2.1}$$

where F_g, $g = 1, ..., G$, is the CDF of data from the gth group and p_1,...,p_G are the unknown mixture proportions. We often assume that the F_g come from some parametric family of distributions; the normal family is most common (McLachlan and Peel, 2000). If we assume that each group follows a normal distribution then (2.1) can be re-written as

$$F(x) = \sum_{g=1}^{G} p_g \Phi(x; \mu_g, \Sigma_g), \tag{2.2}$$

where Φ is the normal CDF and μ_g, Σ_g are the mean vector and covariance matrix of the gth group, $g = 1, ..., G$.

Suppose now that we have a sample, $x_1, ..., x_n$, of iid observations from this distribution F, and we want to construct a rule for assigning observations to clusters. One way to do this is to start with prior probabilities[1] for the mixture proportions, then estimate the parameters of the G normal distributions. We then apply Bayes theorem and assign each observation to the group to which it has the highest posterior probability of belonging. The EM algorithm gives us an efficient way to estimate these parameters and thereby estimate group memberships.

As with other applications of the EM algorithm, we begin by assuming that our data are missing an important component. In this case, we define a latent variable Z_i, $i = 1, ..., n$, a vector in $\{0, 1\}^G$, such that the gth component of Z_i is 1 if observation i comes from group g, and 0 otherwise. This allows us to treat each observation as having been drawn independently from the joint distribution (X_i, Z_i) with Z_i being unobserved in all cases. We can now completely characterize the joint distribution of X and Z. First, the distribution of Z_i is simply multinomial with a single trial and probabilities equal to the mixture proportions $p_1, ..., p_G$. Further, the conditional distribution of X_i given $Z_i = g$ is $\Phi(x; \mu_g, \Sigma_g)$. Note that conditioning on Z_i makes the distribution of X_i much easier to work with. This simplification is key to applying the EM algorithm. The joint distribution of (X_i, Z_i) can be written as,

$$F(x, z) = \prod_{g=1}^{G} \left(p_g \Phi(x; \mu_g, \Sigma_g)\right)^{z^{(g)}} \tag{2.3}$$

where $z^{(g)}$, the gth component of z, is 1 if x belongs to group g and 0 otherwise. This slightly unintuitive formulation of the likelihood is to ensure that its corresponding log-likelihood function is easy to work with. The log-likelihood function corresponding to an observed sample can be written in the following ways with and without Z_i respectively:

$$L_c(\theta; X, Z) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_i^{(g)} \left[\log(p_g) + \log\left(\phi(x_i; \mu_g, \Sigma_g)\right)\right] \tag{2.4}$$

$$L(\theta; X) = \sum_{i=1}^{n} \log\left(\sum_{g=1}^{G} p_g \phi(x_i; \mu_g, \Sigma_g)\right), \tag{2.5}$$

where L_c is referred to as the "complete" log-likelihood, while L is the "incomplete" log-likelihood, ϕ is the normal pdf and θ is the vector of parameters $\mu_1, ..., \mu_G, \Sigma_1, ..., \Sigma_G, p_1, ..., p_G$.

The EM algorithm tries to maximize the log-likelihood function of the data by iterating between an expectation (E)-step, and a minimization (M)-step. The E-step consists of taking the expectation of the complete log-likelihood with respect to the group label variable, Z, for the current values of θ. At the first step, θ is randomly initialized to some

[1] This probability is often estimated in practice, and is therefore not a prior. It is, however, otherwise treated like a prior, and often referred to as one.

value $\theta^{[0]}$ (usually by assigning group membership probabilities to each observation), but in general the expectation at step k is obtained using the value from step $k-1$, $\theta^{[k-1]}$. Mathematically, taking this expectation is equivalent to replacing $z_i^{(g)}$ with the estimated posterior probability of point i belonging to group g; call this $\tau_i^{(g)}(\theta)$. Note that $\tau_i^{(g)}(\theta)$ can be written as

$$\tau_i^{(g)}(\theta) = \frac{p_g \phi(x_i; \mu_g, \Sigma_g)}{F'(x_i; \theta)}, \tag{2.6}$$

where F' is the derivative of the CDF with respect to θ. As mentioned above, in practice θ is set to the parameter values from the previous step of the algorithm. We can now obtain the following closed-form expression for the incomplete likelihood after the kth E-step.

$$L_c^*(\theta; X, \theta^{[k-1]}) = \sum_{i=1}^{n} \sum_{g=1}^{G} \tau_i^{(g)}(\theta^{[k-1]}) \cdot [\log(p_g) + \log(\phi(x_i; \mu_g, \Sigma_g))] \tag{2.7}$$

Note that this expression, L_c^*, is a function of $\theta = \mu_1, ..., \mu_G, \Sigma_1, ..., \Sigma_G, p_1, ..., p_G$, and that $\theta^{[k-1]}$ is treated as fixed. This is because we still need to optimize over θ, whereas $\theta^{[k-1]}$ consists of information we already know from the previous iteration.

The M-step proceeds by maximizing L_c^* for θ. We first obtain $\tilde{\theta}^{[k]}$ by updating the p_g as

$$\hat{p}_g^{[k]} = \sum_{i=1}^{n} \frac{\tau_i^{(g)}(\theta^{[k-1]})}{n} \tag{2.8}$$

This is the average posterior probability across all observations. We now replace θ in L_c^* with $\tilde{\theta}^{[k]} = \mu_1, ..., \mu_G, \Sigma_1, ..., \Sigma_G, \hat{p}_1^{[k]}, ..., \hat{p}_G^{[k]}$ to get

$$\tilde{L} = L(\tilde{\theta}^{[k]}; X, \theta^{[k-1]}) = \sum_{i=1}^{n} \sum_{g=1}^{G} \tau_i^{(g)}(\theta^{[k-1]}) \cdot [\log(\hat{p}_g^{[k]}) + \log(\phi(x_i; \mu_g, \Sigma_g))] \tag{2.9}$$

Finally, we obtain an updated estimate of θ by maximizing $\tilde{L}$ over the remaining unspecified parameters in $\tilde{\theta}^{[k]}$: $\mu_1, ..., \mu_G, \Sigma_1, ..., \Sigma_G$. In general, this maximization can be done either numerically or analytically. For the unconstrained normal model, we get closed form expressions for the $\hat{\mu}_g^{[k]}$ and $\hat{\Sigma}_g^{[k]}$,

$$\hat{\mu}_g^{[k]} = \frac{\sum_{i=1}^{n} \tau_i^{(g)}(\theta^{[k-1]}) \cdot x_i}{\sum_{i=1}^{n} \tau_i^{(g)}(\theta^{[k-1]})} \tag{2.10}$$

$$\hat{\Sigma}_g^{[k]} = \frac{\sum_{i=1}^{n} \tau_i^{(g)}(\theta^{[k-1]}) \cdot (x_i - \hat{\mu}_g)^T (x_i - \hat{\mu}_g)}{\sum_{i=1}^{n} \tau_i^{(g)}(\theta^{[k-1]})} \tag{2.11}$$

Note that these averages and covariance matrices are weighted using the $\tau_i^{(g)}(\theta^{[k-1]})$, not the $\hat{p}_g^{[k]}$. We now update θ to $\theta^{[k]} = \hat{\mu}_1^{[k]}, ..., \hat{\mu}_G^{[k]}, ..., \hat{\Sigma}_1^{[k]}, ..., \hat{\Sigma}_G^{[k]}, \hat{p}_1^{[k]}, ..., \hat{p}_G^{[k]}$.

The algorithm now proceeds by iterating between the E- and M-steps until a suitable stopping criterion is reached (McLachlan and Peel, 2000). The most common criteria are lack of sufficient change in the likelihood and lack of sufficient change in the parameter estimates. Upon convergence, we call the groups "clusters", thus obtaining a clustering solution.

One natural extension to the EM algorithm is called the classification EM, or CEM algorithm (Celeux and Govaert, 1992). This method directly maximizes the complete likelihood, L_c, given in (2.4). That is, when using the CEM algorithm, we treat the Z_i as parameters of interest. We hereafter also refer to this quantity as the "classification likelihood", or C-likelihood, without any change of notation (and take the c subscript to mean complete or classification where appropriate). The CEM algorithm for maximizing this C-likelihood very closely resembles the EM algorithm, and differs from it only by the addition of a C-step between the E- and M-steps. In this C-step, each observation is assigned full membership in the group to which it has the highest probability of belonging. That is, we replace the largest element of $\tau_i^{(g)}(\theta^{[k-1]})$ with 1, and its other elements with 0. Iteration then proceeds as in the EM algorithm, alternating between E-, C- and M-steps until a stopping criterion is met. As with the EM algorithm, these final groups are called clusters.

It is sometimes desirable to assume that the mixture proportions are equal when using the EM or CEM algorithms. While this can be done purely mathematically, a different model exists to motivate imposing this restriction on the C-likelihood function (and therefore on the CEM algorithm). First, note that the C-likelihood function as given in (2.4) arises from sampling observations independently from the mixture distribution, F (Symons, 1981) (i.e., the observations' group memberships follow a multinomial distribution). If we instead sample a fixed number of observations from each component, F_g, then the p_g drop out of our C-likelihood function (Scott and Symons, 1971). This is mathematically equivalent to simply assuming equal mixture proportions (i.e., the p_g no longer have any effect on the optimization problem because they are all equal). Whether the assumption of equal proportions is imposed directly or because of the sampling scheme, we refer to the likelihood and C-likelihood under this assumption as the restricted likelihood and restricted C-likelihood respectively. These functions can be written as follows.

$$L_c^{(R)}(\theta; X, Z) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_i^{(g)} \left[\log\left(\phi(x_i; \mu_g, \Sigma_g)\right)\right] \tag{2.12}$$

$$L^{(R)}(\theta; X) = \sum_{i=1}^{n} \log\left(\sum_{g=1}^{G} \phi(x_i; \mu_g, \Sigma_g)\right) \tag{2.13}$$

Model-Based Clustering and CEM

The methods outlined in the previous section solve the clustering problem. Despite assuming that data come from groups, we do not treat those groups as known. Since we assume a specific model when using the EM and CEM, any clustering method that can be formulated using these frameworks is called a "model-based" clustering algorithm. Celeux and Govaert (1995) identify 14 different model-based clustering algorithms using the CEM algorithm applied to mixture normal data. These different algorithms arise from making various assumptions about the groups' covariance matrices, $\Sigma_1, ..., \Sigma_G$. We use the spectral decomposition of each covariance matrix, $\Sigma_g = \lambda_g D_g A_g D_g^T$ to characterize these 14 models (Banfield and Raftery, 1993). Here D_g is an orthonormal matrix of eigenvectors of Σ_g, A_g is a diagonal matrix of the eigenvalues of Σ_g divided by the largest eigenvalue (so that all entries of A_g are between 0 and 1), and λ_g is a scalar equal to the largest eigenvalue of Σ_g. Note that this decomposition is guaranteed to exist by the Spectral Theorem (Friedberg et al., 1989).

The spectral decomposition of Σ_g is useful for thinking about the geometry of the data in group g. We know that these data come from a multivariate normal distribution, so we can expect their density to form an ellipsoidal shape (Banfield and Raftery, 1993). The components λ_g, D_g and A_g control different aspects of this ellipsoid. First, λ_g controls the size of the ellipsoid. The orientation of the ellipsoid, by which we mean the orientation of its principal axes, is determined by D_g. Finally, A_g determines the relative sizes of the ellipsoid's principal axes, which we call the shape of the ellipsoid. We now use this notation to characterize the 14 different models discussed by Celeux and Govaert (1995).

These 14 models are listed in Table 2.1 using the notation of Celeux and Govaert. Under this scheme the decomposition of Σ_g is written with either g or nothing as a subscript to indicate whether that component is different or the same across groups respectively. Table 2.1 also includes the number of parameters that must be estimated between all the groups' covariance matrices, where d is the dimension of the data (Celeux and Govaert, 1995). In this table, $\beta = d(d+1)/2$ is the number of free parameters in a single, unconstrained, d-dimensional covariance matrix. The last column of Table 2.1 lists the data transformations to which that model is invariant (Celeux and Govaert, 1995). That is, which data transformations have essentially no effect on the fitting procedure (or that the effect is highly predictable, as with translations simply shifting the same problem some number of units in a particular direction). Knowing the invariance properties of these models helps us later to understand a limitation of Yang's algorithm (Yang, 2013). The first eight models are invariant to any affine transformation (e.g., rotations, translations or re-scaling)[2]. The next

[2] Celeux and Govaert (1995) state that these models are invariant to linear transformations. To see that they are also translation invariant, note that estimation of the Σ_g is translation invariant, and that estimation of the μ_g responds to translations exactly as we would expect. Finally, the assignment of observations to groups is based on the Mahalanobis distance between an observation and the centroid of each group, and Mahalanobis distances are translation invariant (Haasdonk and Pękalska, 2009).

four models that assume diagonal covariance matrices are invariant to axis rescaling; that is, linear transformations that correspond to a diagonal matrix. Finally, the last two models that assume spherical covariance matrices are invariant to isometric transformations (i.e., transformations that preserve distances between points), such as rotations and translations. See the lecture notes by Conrad (2016) for more details on isometric transformations. Of particular interest for our simulations (see Section 3) is Conrad's Theorem 4.1, which shows that rotations about the origin in $\mathbb{R}^n$ are linear transformations. Therefore, we can conclude that the first eight models we have discussed are isotropic, in addition to the last two. The middle four models are not isotropic however, because rotations cannot be expressed a special case of axis re-scaling.

The first eight models in our list can be constructed by requiring that various components of the spectral decomposition of Σ_g be the same or different between groups. As discussed above, this changes the geometry of our data. For example, requiring that all D_g are equal to one common matrix, D, gives us groups whose ellipses have the same orientation. There are three components to the spectral decomposition, so allowing each to be either the same or different across groups gives us $2^3 = 8$ different models ranging from the most general, with no common parameters, to the most specific, where each group has the same covariance matrix. Our second group of models constrains D_g to be the identity matrix, which results in Σ_g being diagonal. This gives us an ellipsoid whose principal axes are oriented with the coordinate axes. We now have two components that can be either the same or different across groups (A and λ), which gives us four more models. Finally, we can further constrain our model so that the A_g are all equal to the identity matrix. This gives us ellipsoids that are actually hyperspheres. In this setting, we are left with a single parameter to either vary or hold constant across groups (λ), which gives us our final two models. Note that we do not include D_g in our model after constraining A_g to be the identity matrix, because orientation is meaningless when contours of the distribution form a hypersphere.

Using the 14 covariance structures we just identified, together with the two algorithms (EM or CEM) discussed in the previous section and the two likelihood forms (restricted or unrestricted), we are now able to express many non-hierarchical clustering algorithms within our framework. For example, k-means is the CEM algorithm with restricted C-likelihood and the $[\lambda I]$ covariance structure (Celeux and Govaert, 1992). Similarly, the soft k-means algorithm mentioned briefly in Section 2.1 is identical to regular k-means, but uses the EM algorithm. Another common approach to characterize clustering algorithms is to express them as optimization algorithms with a particular objective function (Windham, 1987). In their list of 14 models, Celeux and Govaert (1995) also include the objective functions that they correspond to when viewed as optimization algorithms (provided that an objective function exists).

Table 2.1: The 14 normal clustering models, total number of parameters in their covariance matrices and their invariance properties.

Model Name	Number of Parameters	Invariant Under
$[\lambda DAD^T]$	β	Affine Transformations
$[\lambda_g DAD^T]$	$\beta + G - 1$	Affine Transformations
$[\lambda DA_g D^T]$	$\beta + (G-1)(d-1)$	Affine Transformations
$[\lambda_g DA_g D^T]$	$\beta + (G-1)d$	Affine Transformations
$[\lambda D_g AD_g^T]$	$G\beta - (G-1)d$	Affine Transformations
$[\lambda D_g A_g D_g^T]$	$G\beta - (G-1)$	Affine Transformations
$[\lambda_g D_g AD_g^T]$	$G\beta - (G-1)(d-1)$	Affine Transformations
$[\lambda_g D_g A_g D_g^T]$	$G\beta$	Affine Transformations
$[\lambda A]$	$d+1$	Axis Scaling
$[\lambda_g A]$	$G+d$	Axis Scaling
$[\lambda A_g]$	$Gd+1$	Axis Scaling
$[\lambda_g A_g]$	$G(d+1)$	Axis Scaling
$[\lambda I]$	1	Isometric Transformations
$[\lambda_g I]$	G	Isometric Transformations

2.1.2 Convergence and Asymptotic Properties

The first question we must address here is whether or not the EM and CEM algorithms terminate. Once this has been established, we can investigate what properties this terminal value has, both in terms of the value of the likelihood function and of the parameter estimates.

The EM algorithm turns out to be easier to discuss. Given regularity conditions, it can be shown that the likelihood function converges monotonically to a stationary value under the EM algorithm (Wu, 1983; McLachlan and Krishnan, 2008). Further, depending on the strength of regularity conditions we are willing to assume, it can be shown that the parameter estimates from the EM algorithm converge to local or even global maximizers of the likelihood function. See Wu (1983) for details. Unfortunately, the condition required for guaranteed convergence to a global maximum is rarely met in practice, so it is often recommended to re-run the EM algorithm several times with different initial conditions and choose the solution with the highest likelihood (Wu, 1983; McLachlan and Krishnan, 2008). While not a guarantee of global optimality, this does help reduce the chance of choosing a local optimum.

Provided that sufficiently strong assumptions on the form of the likelihood are met, we are also able to guarantee that parameter estimates from the EM algorithm converge to the MLEs for their respective parameters. They therefore inherit all the properties of MLEs such as consistency, equivariance and asymptotic efficiency (see, for example, Wasserman, 2004).

The CEM algorithm is similarly guaranteed to terminate at a fixed point in terms of the C-likelihood (Celeux and Govaert, 1992). This requires only very mild assumptions on the form of the likelihood function because there are only finitely many partitions of the observed data into clusters.

Unfortunately, parameter estimates obtained from the CEM algorithm are not as well behaved as those from the EM algorithm. Given some regularity conditions, parameter estimates from the CEM algorithm converge to maximizers of the C-likelihood (Celeux and Govaert, 1992), which are not necessarily MLEs. In fact, maximizing the C-likelihood leads to estimators that are not necessarily consistent for the parameters they are estimating (Marriott, 1975; Bryant and Williamson, 1978). Another limitation of the CEM algorithm is that, as with the EM algorithm, reasonable assumptions only allow us to guarantee convergence to a local maximum. The recommended solution to this is, as above, to re-run the algorithm several times with different initializations and choose the solution with the highest likelihood.

2.2 Discrimination

2.2.1 Methods

The problem of discrimination begins with a sample of n observations in $\mathbb{R}^d$, x_1,...,x_n, that are a priori divided into G labeled groups (Hastie et al., 2011). The goal is then to partition d-space into G disjoint regions and assign each region a group such that a new observation in each region is most likely to belong to that region's group (Hand, 1981). In general, these groups can have any distribution, but the most common methods, LDA (Linear Discriminant Analysis) and QDA (Quadratic Discriminant Analysis), assume that the groups are normally distributed with different means. Further, LDA assumes that the groups have the same covariance matrix, while QDA imposes no such constraint. Note that the covariance structures assumed for LDA and QDA correspond to rows one and eight respectively in Table 2.1. More generally, we can perform a DA with any of the 14 covariance assumptions listed in this table. This possibility is investigated by Bensmail and Celeux (1996). All discrimination methods we discuss assume one of these 14 models.

Once a model has been chosen, the typical next step is to estimate the mean, covariance matrix and "prior probability" for each group using maximum likelihood (we still call it a prior even though it is estimated from data). The parameters in a group are estimated using only the points belonging to that group. Estimators of the mean, μ_g, and prior probability,

p_g, for each group do not depend on our covariance assumption, and are as follows,

$$\hat{\mu}_g = \sum_{\mathcal{G}(x_i)=g} \frac{x_i}{n_g} \tag{2.14}$$

$$\hat{p}_g = \frac{n_g}{n} \tag{2.15}$$

where $\mathcal{G}(x_i) = g$ when x_i belongs to group g, and n_g is the number of observations in group g.

Estimating the covariance matrix can be more challenging, as some of the models in Table 2.1 do not have closed form covariance MLEs. Bensmail and Celeux (1996) list which models have closed form estimators and which do not. If no closed form exists, an iterative numerical method can be used to obtain an estimate of the covariance matrix. Fortunately, both LDA and QDA have closed form covariance estimates. We begin with the estimator of the covariance matrix for each group, Σ_g, in QDA.

$$\hat{\Sigma}_g = \sum_{i:\mathcal{G}(x_i)=g} \frac{(x_i - \hat{\mu}_g)^T (x_i - \hat{\mu}_g)}{n_g - 1} \tag{2.16}$$

The estimate of the common covariance matrix in LDA, Σ, is as follows.

$$\hat{\Sigma} = \sum_{g=1}^{G} \frac{n_g - 1}{n - G} \hat{\Sigma}_g \tag{2.17}$$

Once we have estimated the parameters of the distribution in each group, we put these together to form our decision rule. The idea is to assign a point, x_0, to group g if the "posterior probability" (i.e., likelihood times "prior") of x_0 belonging to group g is larger than it is for any other group. Mathematically, we formulate this as follows,

$$\hat{\mathcal{G}}(x_0) = \arg\max_g \hat{p}_g \Phi(x_0; \hat{\mu}_g, \hat{\Sigma}_g) \tag{2.18}$$

where for LDA, we replace $\hat{\Sigma}_g$ with $\hat{\Sigma}$.

2.2.2 Asymptotic Properties

The goal of any discrimination method is to emulate the so called "Bayes rule". This rule is defined as a method that assigns every point in the sample space to the group to which it has the highest true probability of belonging (Hastie et al., 2011). Therefore the Bayes rule is, by construction, the best possible discrimination function. If this discriminator is then used for classification, the probability that it misclassifies a new observation, its "misclassification rate", is called the "Bayes rate", and this is the best misclassification rate achievable by a classifier. Our goal with any discrimination method is therefore to emulate the Bayes rule as closely as possible.

The types of DA that we present above are based on likelihood estimation. Therefore, the parameter estimates inherit the properties associated with MLEs. In particular, since $\hat{p}_g, \hat{\mu}_g$ and $\hat{\Sigma}_g$ are all MLEs, our discriminator, $\hat{\mathcal{G}}$, is also an MLE for the label of the group with the largest posterior probability. This result follows from the equivariance property of MLEs (Casella and Berger, 2002). We can therefore conclude that, provided the model is correctly specified, the discriminator obtained from this method is a consistent estimator of the Bayes rule (Wasserman, 2004).

The above argument suggests that a less restrictive class of models may be preferable to a more restrictive one, because fewer restrictions means that the model is less likely to be misspecified. While this is true asymptotically, in finite samples parameter estimation adds variability to the predicted values (Hastie et al., 2011). Choosing a model becomes a classic example of the bias-variance trade-off (Hastie et al., 2011). While QDA uses the most general model possible and thus gives a discriminator that asymptotically unbiased for the Bayes rule for the largest class of populations, it may require an extremely large dataset to estimate with any precision. The discriminator obtained from LDA can be estimated with more precision (it reduces the number of covariance parameters to estimate by a factor of G), so it is in general less variable than QDA, but it may not be asymptotically unbiased. Determining which method performs better in a particular situation requires empirically evaluating both methods.

2.3 Comparing Methods

All methods discussed in this chapter are designed to solve very similar problems: given a dataset where observations are known to belong to different groups, construct a rule for partitioning the sample space into clusters (or regions) that are as similar as possible to the population groups. It is therefore natural to wonder which method is best able to solve this problem. A common way to frame this question is in terms of the related problem of classification (Celeux and Govaert, 1993, 1995; Flury et al., 1994). Although clustering algorithms are not explicitly intended to perform classification, this is a natural extension of their intended use, and a way in which they are often compared.

In order to evaluate the methods we have described in this chapter, we study their misclassification rates. That is, for each method, we investigate the probability that it assigns a new observation to the wrong group. If a method constructs a reasonable partition of the sample space then its error rate should be close to the Bayes rate and, inversely, a poor partition should lead to a much higher misclassification rate.

There are three factors by which we can characterize the methods discussed above. The first is which approach is used to estimate the parameters of the model: EM, CEM, or DA.

Note that the first two of these do not require that group labels are available, while the third does. The second is what assumptions we make about the covariance structure of the population distribution when we perform our analysis. The third is whether we use the restricted or unrestricted likelihood. That is, whether we assume that the mixture proportions in the population are equal. Many authors have investigated these factors in various ways. Castelli and Cover (1995, 1996) investigate some difficulties that arise when working with data where group labels are only available for some observations (a.k.a. “semi-supervised” learning). Celeux and Govaert (1993) compare the EM and CEM algorithms under several different model specifications (including restricted and un-restricted likelihood) and provide some insights into when one approach can be expected to outperform the others. In a later paper, Celeux and Govaert (1995) investigate all 14 covariance structures when used with the CEM algorithm and the restricted likelihood. Flury et al. (1994) perform a similar investigation on DA, but with a more limited scope. Finally, Bensmail and Celeux (1996) extend this investigation of DA to include all 14 covariance structures.

One common theme between these investigations is their empirical nature. It is difficult to obtain analytic results about the performance of these methods. We therefore use a simulation study to compare the performance of EM, CEM and DA under various model specifications. We also include Yang’s supervised clustering as an alternative model.

Chapter 3

Simulation Studies

We carry out two simulation studies to analyze the methods discussed in Chapter 2. Both involve comparing analysis procedures on many different datasets. The first simulation compares DA to the EM algorithm. The CEM algorithm is excluded from this simulation due to computational constraints (See Section 4.2). Our second simulation compares the k-means algorithm to its EM and DA counterparts. That is, we compare DA and the EM and CEM algorithms only when assuming that the groups' covariance matrices are equal and proportional to the identity matrix and that the mixture proportions are equal. We refer to our two simulation studies as "EM vs DA", and "k-means comparison", respectively.

3.1 EM vs DA

3.1.1 Overview

The goal of this simulation is to investigate the analysis of data from mixture normal distributions using the four different techniques discussed in the previous chapter (EM, CEM, DA and Yang's). Specifically, we want to evaluate how well these methods work as classifiers. That is, our goal is to identify what factors affect the expected misclassification rate (i.e., probability of misclassification) of these three methods, and when we can expect one to outperform the others. Note that both DA and Yang's pre-processing procedure are forms of supervised learning, so we are particularly interested in comparing their performance to that of the other methods.

The form of our simulation follows a split-plot design (Milliken and Johnson, 1992). Data are generated with a number of settings; these are the whole-plot factors. For each whole-plot (combination of settings for generating data), we can generate a number of individual datasets; these are the sub-plots. Each dataset is analyzed using methods outlined in the previous chapter with various settings; these settings are the sub-plot factors. Each dataset in our simulation consists of n observations from a mixture of two multivariate normal distributions in $\mathbb{R}^d$. The settings we vary on the datasets are the difference between the means, the mixture proportions and the structure of the groups' covariance matrices. We also consider different dimensions, d, and different values of the sample size, n. Note that, each time we generate a new dataset, we also generate new covariance matrices for its groups, subject to the constraint being applied. This allows us to generalize our inference over the space of covariance matrices satisfying the specified properties.

We compare the various analysis procedures discussed in the previous chapter by running them all on each dataset and estimating their error rates. The different procedures that we are interested in make up the sub-plot factors of our split-plot design. The specific factors we are interested in are (1) which method we use (EM or DA); (2) whether we assume equal mixture proportions; (3) what restrictions we place on the covariance matrices of the groups and; (4) if we assume equal spherical covariance matrices, whether we apply Yang's pre-processing method. For convenience, we treat Yang's pre-processing as a fourth covariance structure. Any model that uses Yang's pre-processing procedure or that is fit using DA is considered "supervised", and any that do not are considered "unsupervised". We discuss all these settings in more detail in Section 3.1.2.

Once a combination of whole-plot treatments has been selected and a dataset has been generated, we fit all the models (i.e., all level combinations of the analysis factors). Once all models have been fit, we generate 1000 new observations from the data distribution and use our models to assign them to groups. We then compute the number of new observations that are assigned to the wrong group by each model, which is an estimate of that model's misclassification rate[1]. This is the response variable in our experiment. We then generate 4 more training and test datasets and repeat this procedure on each of them, giving us misclassification rate estimates for each model on 5 different datasets. That is, we generate 5 replications for each combination of whole-plot factors, and split each of these into a separate sub-plot for each combination of the sub-plot factors. More formally, the whole-plot experimental units are datasets, and the sub-plot experimental units are individual analyses.

This simulation is carried out in `R` (R Core Team, 2016) and analyzed using `SAS`® software (SAS Institute Inc., 2013). All versions of DA and the EM algorithm are implemented using the `MclustDA` and `Mclust` functions in the `mclust` package (Biernacki et al., 2006). The logistic regression method developed by Heinze and Schemper (2002) is implemented using the `logistf` function in the eponymous package (Heinze et al., 2013). The simulation itself is carried out in parallel using the `plyr` and `doParallel` packages (Wickham, 2011; Revolution Analytics and Weston, 2015). Analysis of the misclassification rates is performed using the `MIXED` procedure, and all plots are constructed using the `GPLOT` procedure in `SAS`.

[1] When an entirely unsupervised method is used, it is not immediately obvious how to assign group labels to the clusters. This difficulty is investigated in detail in a series of papers by Castelli and Cover (1995, 1996). We avoid the problem entirely by assigning group labels such that the misclassification rate is minimized.

An odd phenomenon occurs in a small number of generated datasets that have a substantial outlier. In rare cases, this outlier is so extreme that it is assigned to its own cluster, while the rest of the observations form the other cluster. This makes estimating covariance matrices impossible. In cases where this occurs, we replace the dataset with a new one and re-fit all models. We then repeat the procedure as necessary until both clusters contain at least two observations. This scenario occurs in less than 0.3% of cases (89 out of 32400 model fits). These replacements occur mostly when we do not assume equal mixture proportions or when the EM algorithm is used, and almost never when no restrictions are placed on the covariance matrix.

3.1.2 Parameter Settings

Many of the parameter values we consider are chosen to match those in one of the simulations used by Celeux and Govaert (1993) to compare results from the EM and CEM algorithms. Let μ_g and Σ_g be the mean and covariance matrix of group $g = 1, 2$, and let p and $q = 1-p$ be the mixture proportions. Then $X_1, ..., X_n \overset{iid}{\sim} F_X$, where $F_X(x) = p \cdot \Phi(x; \mu_1, \Sigma_1) + q \cdot \Phi(x; \mu_2, \Sigma_2)$. We vary the μ_g's by setting the Mahalanobis distance between them[2] to be 1, 2, 3, 4 or 5. We use Mahalanobis distance rather than Euclidean distance because we use a different covariance matrix each time we generate a new dataset, and the Mahalanobis distance is invariant to this change. The Mahalanobis distance is also listed as a multivariate generalization for the coefficient of variation by Aerts et al. (2015), which is a reasonable measure of the inverse "signal-to-noise ratio". Using Mahalanobis distance to measure the signal-to-noise ratio is particularly appropriate here since it is the only measure listed by Aerts that is invariant to affine transformations of the data (therefore it is not affected when we standardize the variables or apply Yang's pre-processing procedure) (Haasdonk and Pękalska, 2009). It is not clear how to define the Mahalanobis distance between the groups' means when their covariance matrices are different. We therefore use the Mahalanobis distance relative to the covariance matrix of group 2.

The first mixture proportion, p, takes the values $0.25, 0.35$ and 0.5. The dimension, d, takes the values $2, 4$ and 6. This differs from the values in Celeux and Govaert (1, 2 and 4), and is chosen so that the simulation does not become too large to compute efficiently. The sample size, n, takes the values $200, 400$ and 600. These also differ from the values used by Celeux and Govaert (20, 40, 100 and 200), and are chosen to ensure that there is enough data to fit our models using the `R` package `Mclust` (Fraley et al., 2012)[3]. The covariance matrices are either equal and proportional to the identity matrix, unconstrained but equal, or completely unconstrained, and are generated randomly with a determinant of 1. This

[2] The Mahalanobis distance between two points relative to some covariance matrix, Σ, is $d_\Sigma(x, y) = (x-y)^T \Sigma^{-1} (x-y)$ (Wasserman, 2004).

[3] The number of observations required to fit models using `Mclust` is variable, so a sufficiently large sample size is required to ensure all models can be fit.

Table 3.1: Factors for the EM vs DA simulation, with their numbers of levels and whether they are applied to data generation or data analysis.

Factor	Levels	Type
Mean Difference	5	Generation
Mixture Proportion	3	Generation
Dimension	3	Generation
Sample Size	3	Generation
Covariance Structure	3	Generation
Assumed Covariance Structure	4	Analysis
Assume Equal Mixture	2	Analysis
Method	2	Analysis

is chosen to ensure that the volume of the groups' covariance ellipsoids (alternatively, the volume of the parallelepiped generated by the columns of each group's covariance matrix) are all 1, regardless of what constraints are imposed (Peng, 2007). Note that this still requires that we estimate the λ parameter for each covariance structure (see Section 2.1.1) because constraining the determinants of two matrices to be the same is not equivalent to constraining their largest eigenvalues to be the same.

The assumed covariance structures for our analyses are the same as the three structures used to generate data. Further, if the covariance matrices are assumed to both be proportional to the identity matrix, then Yang's pre-processing procedure is either applied or not. This procedure is not considered for other covariance structures because their fitting methods are invariant to linear transformations of the data (See Table 2.1). The mixture proportions can either be assumed equal or not. Finally, the fitting method is either EM or DA. Recall that the EM algorithm is often fit multiple times and the best result selected (to avoid locally optimal solutions, see Section 2.1.2). We therefore fit the model 20 or 45 times and select the best one (preliminary analyses suggest that 20 is sufficient for the likelihood to stabilize when the Mahalanobis distance between the means is at least 2, and 45 is required when this distance is 1). All factors that we consider are listed in Table 3.1, along with their number of levels and whether they pertain to data generation or data analysis.

3.1.3 Analysis

We begin this section by describing our statistical analysis, and identifying which effects have a significant impact on the misclassification rate. We then list the estimated misclassification rate for each level of the analysis main effects, and present plots showing the effects for some interesting two-factor interactions. Finally, we discuss several contrasts that correspond to interesting questions about the data.

The results of this simulation are analyzed as a split-plot design (Littell et al., 2006; Milliken and Johnson, 1992). The data generation process corresponds to whole-plots, with data variables (i.e., mean difference, mixture proportion, dimension, sample size and covariance structure) being whole-plot factors. The whole-plot experimental units are datasets. The analysis and error rate estimation process corresponds to sub-plots, with the analysis variables (i.e., assumed covariance structure, assumed equal mixture and fitting method) being sub-plot factors. Our response variable is the misclassification rate of a particular analysis on a single dataset. Note that this is a proportion, so we use the appropriate variance-stabilizing transformation, $\arcsin(\sqrt{Y})$ (Kuehl, 2000). We are therefore implicitly assuming that the transformed error rates follow normal distributions with constant variance across all levels of the predictor variables.

We fit a mixed-effects linear model to predict the error rate using the whole- and sub-plot factors, along with two- and three-way interaction terms within and between plot levels. Higher order interactions are excluded both for ease of computation (large mixed-effects models take a long time to fit) and because high order interaction terms are difficult to interpret. This restriction of third-oder interactions ensures that we have a large number of observations (i.e. degrees of freedom) with which to carry out each test. Specifically, the estimate for each effect is averaged across the others, which gives a large number of observations with which to estimate each mean. Further, this large number of observations allows us to invoke the De Moivre-Laplace Theorem (Durrett, 2013) to justify our assumption of normality. Assuming normality after the variance stabilizing transformation is actually more accurate on small samples than simply assuming that the data are normally distributed (Bromiley and Thacker, 2002).

Of particular interest to us are the variables that pertain to data analysis because our goal is to provide recommendations for analysis, and these are the only variables that an analyst has control over. Factors and interactions that are significant at the $\alpha = 0.05$ level are listed in Tables 3.2-3.5. A simple Bonferroni correction for the number of tests being conducted (Wasserman, 2004) suggests using $\alpha = 5.4 \cdot 10^{-4}$. We indicate effects that are also significant at this level by adding an exclamation point. Table 3.2 contains terms that only pertain to data generation, while Tables 3.3, 3.4 and 3.5 contain respectively terms that include whether the mixture proportions are assumed equal, the assumed covariance structure and the fitting method. Note that some effects occur in more than one of these tables, since there are significant interactions between analysis variables.

Next, we investigate the estimated misclassification rate for different levels of these significant effects. Evaluating all pairwise differences for each effect requires over 1500 tests, so we give only interesting highlights here. We focus on the sub-plot factors (i.e., assumed equal mixture, assumed covariance structure and method) because these are the variables that an analyst sets, and our goal is to provide recommendations for the procedures under investigation.

We give estimates of the misclassification rate after back-transforming (this is, on the error rate scale rather than the transformed-error rate scale) at each level of the analysis

Table 3.2: Data generation effects in the EM vs DA simulation that are significant at the $\alpha = 0.05$ level. Effects that are are also significant at the Bonferroni corrected level are indicated with an exclamation point (see text).

Source	
Mean Difference	!
Mixture Proportion	!
Mean Difference*Mixture Proportion	!
Dimension	!
Mean Difference*Dimension	!
True Covariance Structure	!
Mean Difference*True Covariance Structure	!
Mixture Proportion*True Covariance Structure	!
Dimension*True Covariance Structure	!
Mean Difference*Dimension*True Covariance Structure	!
Mixture Proportion*Dimension*True Covariance Structure	
Mean Difference*Sample Size	!
Mixture Proportion*Sample Size	
Dimension*Sample Size	

Table 3.3: Effects in the EM vs DA simulation containing “Assumed Equal Mixture”, that are significant at the $\alpha = 0.05$ level. Effects that are are also significant at the Bonferroni corrected level are indicated with an exclamation point (see text).

Source	
Assumed Equal Mixture	!
Assumed Equal Mixture*Assumed Covariance Structure	!
Assumed Equal Mixture*Method	!
Assumed Equal Mixture*Assumed Covariance Structure*Method	!
Mean Difference*Assumed Equal Mixture	!
Mean Difference*Assumed Equal Mixture*Assumed Covariance Structure	!
Mean Difference*Assumed Equal Mixture*Method	!
Mixture Proportion*Assumed Equal Mixture	!
Mixture Proportion*Assumed Equal Mixture*Assumed Covariance Structure	!
Mixture Proportion*Assumed Equal Mixture*Method	!
Mean Difference*Mixture Proportion*Assumed Equal Mixture	!
Dimension*Assumed Equal Mixture	!
Dimension*Assumed Equal Mixture*Assumed Covariance Structure	!
Dimension*Assumed Equal Mixture*Method	!
Mixture Proportion*Dimension*Assumed Equal Mixture	
True Covariance Structure*Assumed Equal Mixture*Assumed Covariance Structure	!
Mixture Proportion*True Covariance Structure*Assumed Equal Mixture	!
Dimension*True Covariance Structure*Assumed Equal Mixture	
Sample Size*Assumed Equal Mixture*Assumed Covariance Structure	!
Mixture Proportion*Sample Size*Assumed Equal Mixture	

Table 3.4: Effects in the EM vs DA simulation containing “Assumed Covariance Structure”, that are significant at the $\alpha = 0.05$ level. Effects that are are also significant at the Bonferroni corrected level are indicated with an exclamation point (see text).

Source	
Assumed Covariance Structure	!
Assumed Equal Mixture*Assumed Covariance Structure	!
Assumed Covariance Structure*Method	!
Assumed Equal Mixture*Assumed Covariance Structure*Method	!
Mean Difference*Assumed Covariance Structure	!
Mean Difference*Assumed Equal Mixture*Assumed Covariance Structure	!
Mean Difference*Assumed Covariance Structure*Method	!
Mixture Proportion*Assumed Covariance Structure	!
Mixture Proportion*Assumed Equal Mixture*Assumed Covariance Structure	!
Mixture Proportion*Assumed Covariance Structure*Method	!
Mean Difference*Mixture Proportion*Assumed Covariance Structure	!
Dimension*Assumed Covariance Structure	!
Dimension*Assumed Equal Mixture*Assumed Covariance Structure	!
Dimension*Assumed Covariance Structure*Method	!
Mean Difference*Dimension*Assumed Covariance Structure	!
Mixture Proportion*Dimension*Assumed Covariance Structure	!
True Covariance Structure*Assumed Covariance Structure	!
True Covariance Structure*Assumed Equal Mixture*Assumed Covariance Structure	!
True Covariance Structure*Assumed Covariance Structure*Method	!
Mean Difference*True Covariance Structure*Assumed Covariance Structure	!
Mixture Proportion*True Covariance Structure*Assumed Covariance Structure	!
Dimension*True Covariance Structure*Assumed Covariance Structure	!
Sample Size*Assumed Covariance Structure	!
Sample Size*Assumed Equal Mixture*Assumed Covariance Structure	!
Sample Size*Assumed Covariance Structure*Method	!
Mean Difference*Sample Size*Assumed Covariance Structure	!
Mixture Proportion*Sample Size*Assumed Covariance Structure	
Dimension*Sample Size*Assumed Covariance Structure	!

Table 3.5: Effects in the EM vs DA simulation containing "Method", that are significant at the $\alpha = 0.05$ level. Effects that are are also significant at the Bonferroni corrected level are indicated with an exclamation point (see text).

Source	
Method	!
Assumed Equal Mixture*Method	!
Assumed Covariance Structure*Method	!
Assumed Equal Mixture*Assumed Covariance Structure*Method	!
Mean Difference*Method	!
Mean Difference*Assumed Equal Mixture*Method	!
Mean Difference*Assumed Covariance Structure*Method	!
Mixture Proportion*Method	!
Mixture Proportion*Assumed Equal Mixture*Method	!
Mixture Proportion*Assumed Covariance Structure*Method	!
Mean Difference*Mixture Proportion*Method	!
Dimension*Method	!
Dimension*Assumed Equal Mixture*Method	!
Dimension*Assumed Covariance Structure*Method	!
Mean Difference*Dimension*Method	!
Mixture Proportion*Dimension*Method	
True Covariance Structure*Method	!
Mean Difference*True Covariance Structure*Method	!
Mixture Proportion*True Covariance Structure*Method	!
Dimension*True Covariance Structure*Method	!
Sample Size*Method	!
Sample Size*Assumed Covariance Structure*Method	!
Mean Difference*Sample Size*Method	!
Dimension*Sample Size*Method	
True Covariance Structure*Sample Size*Method	

Table 3.6: Estimated error rate for whether or not equal mixture proportions are assumed in the EM vs DA simulation.

Assumed Equal Mixture	Estimate
No	0.1381
Yes	0.1432

Table 3.7: Estimated error rate for all assumed covariance structures in the EM vs DA simulation. The "spherical" structures correspond to all covariance matrices being equal and proportional to the identity matrix, and "supervised spherical" corresponds to this structure with Yang's pre-processing.

Assumed Covariance	Estimate
Different	0.1223
Equal	0.1814
Spherical	0.1446
Supervised Spherical	0.1189

variable main effects in Tables 3.6-3.8. All the differences between factor levels are significant, even after applying Tukey's correction for the number of tests being carried out (Milliken and Johnson, 1992). This is more important here than when testing for an entire effect, because many more tests are required for all pairwise differences. We then present plots for some interesting two-way interactions in Figures 3.1-3.5. Note that the covariance structures "Hetero", "Homo", "Sph" and "Sup_Sph" correspond respectively to completely unconstrained covariance matrices (heteroscedastic), equal but otherwise unconstrained covariance matrices (homoscedastic), all covariance matrices equal and proportional to the identity matrix (i.e., spherical groups) and all covariance matrices equal and proportional to the identity matrix after Yang's pre-processing procedure has been applied (i.e., spherical groups with supervision).

Next, we give some relevant contrasts, along with their estimates, standard errors and p-values, in Table 3.9. Note that the estimates and standard errors of these contrasts are on the transformed scale (i.e., $\arcsin(\sqrt{Y})$ scale), and that smaller (i.e., more negative) values correspond to lower misclassification rates in the direction suggested by the contrast description. All contrasts are found to be significantly different from zero at the $\alpha = 0.05$ level. We explain the meaning of each contrast here, and interpret their significance in Section 4.1. Note that contrasts are divided into two types. One type compares the means at different combinations of effects. The other type compares the mean of one effect at different levels of another. The first six contrasts are of the first type, and the last four are of the second type.

The first contrast, 'DA vs EM without pre-processing", evaluates whether DA outperforms EM when one of the unsupervised covariance structures is assumed (i.e., not Yang's

Table 3.8: Estimated error rate for all methods in the EM vs DA simulation.

Method	Estimate
DA	0.0906
EM	0.2035

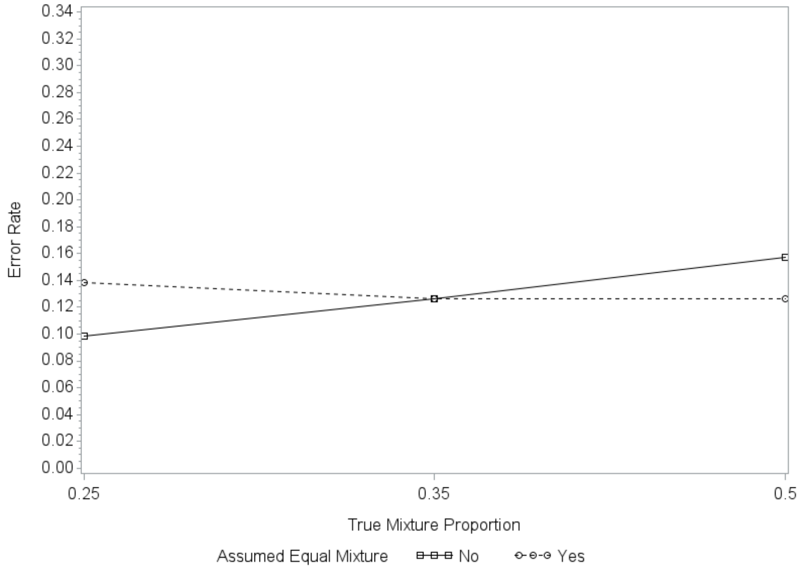


Figure 3.1: Plot of the interaction effect between true mixture proportion and assuming equal mixture proportions in the EM vs DA simulation. See text for details.

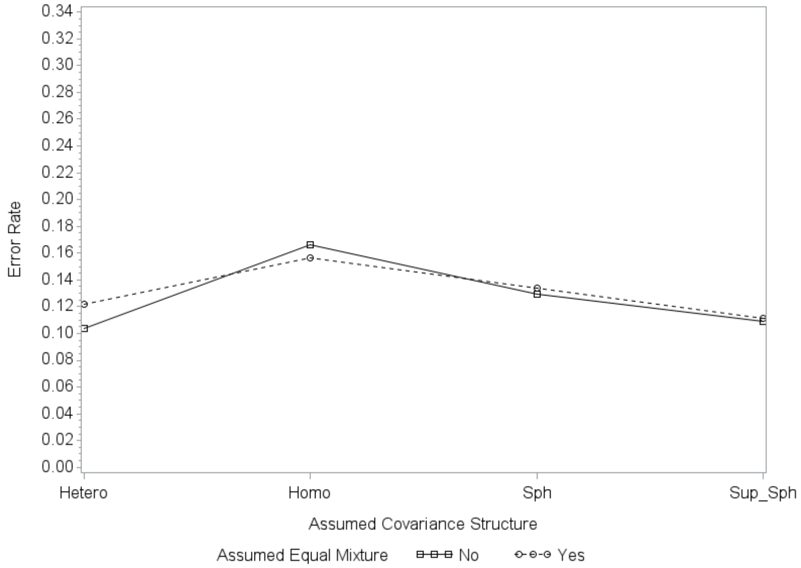


Figure 3.2: Plot of the interaction effect between assumed covariance structure and assuming equal mixture proportions in the EM vs DA simulation. See text for details.

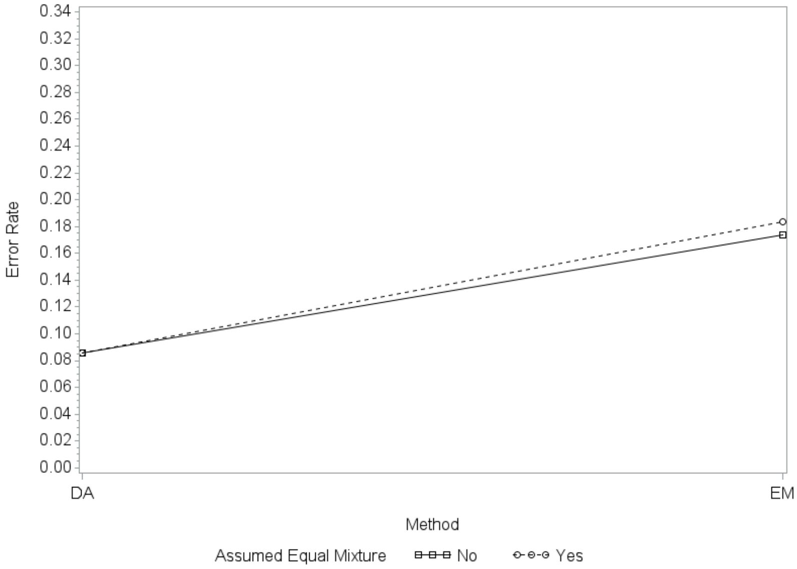


Figure 3.3: Plot of the interaction effect between fitting method and assuming equal mixture proportions in the EM vs DA simulation. See text for details.

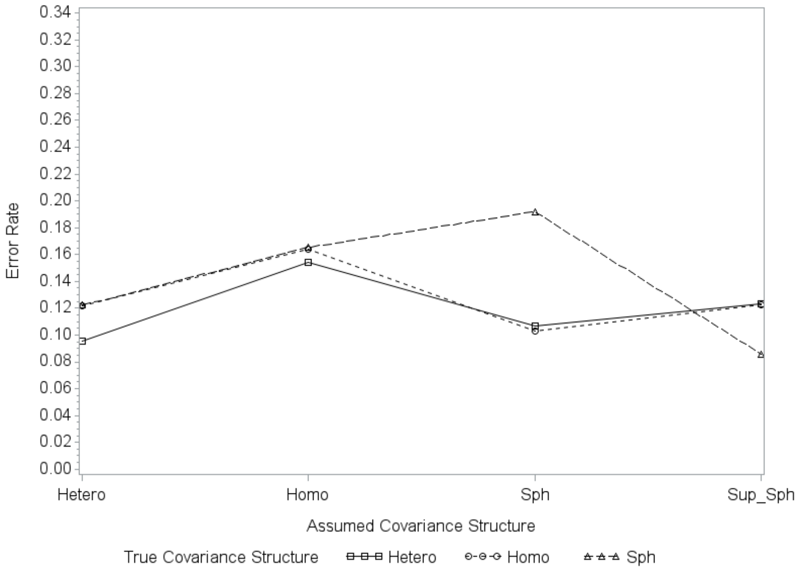


Figure 3.4: Plot of the interaction effect between assumed covariance structure and true covariance structure in the EM vs DA simulation. See text for details.

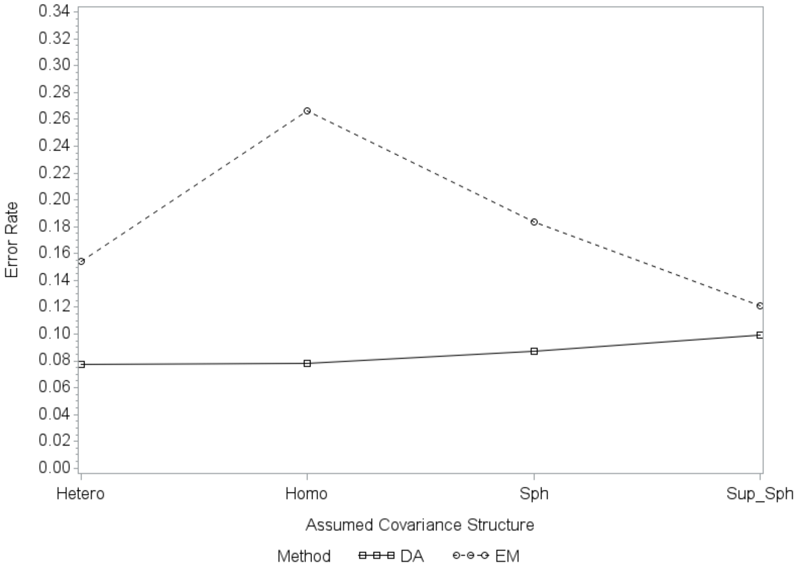


Figure 3.5: Plot of the interaction effect between assumed covariance structure and fitting method in the EM vs DA simulation. See text for details.

Table 3.9: Interesting contrasts for the EM vs DA simulation. See text for a detailed description of the meaning of each contrast.

Label	Estimate	Standard Error	P-Value
DA vs EM without pre-processing	-0.1746	0.001057	<.0001
Pre-processing vs not for EM	-0.1078	0.001495	<.0001
Pre-processing vs spherical for EM	-0.08751	0.001831	<.0001
DA and pre-processing vs one without the other	-0.00091	0.001495	0.5417
DA vs pre-processing	-0.0668	0.001495	<.0001
DA vs both DA and pre-processing	-0.03249	0.001495	<.0001
Effect of DA with vs without pre-processing	0.1403	0.002114	<.0001
Effect of DA for unequal vs equal covariances	0.1371	0.00259	<.0001
Effect of DA for equal vs equal and spherical	-0.1162	0.00259	<.0001
Effect of pre-processing for DA vs EM	0.1403	0.002114	<.0001

pre-processing). The second contrast, “Pre-processing vs not for EM”, evaluates whether using Yang’s pre-processing procedure outperforms the other assumed covariance structures when the EM algorithm is used (i.e., when the fitting method is unsupervised). The third contrast, “Pre-processing vs spherical for EM”, evaluates whether Yang’s pre-processing procedure outperforms the corresponding unsupervised assumed covariance structure (i.e., all equal and proportional to the identity matrix) when the EM algorithm is used. The fourth contrast, “DA and pre-processing vs one without the other”, evaluates whether using DA and Yang’s pre-processing outperforms the average of Yang’s pre-processing with the EM algorithm, and DA with the unsupervised covariance structures. The fifth contrast, “DA vs pre-processing”, evaluates whether using DA alone gives a lower error rate than using pre-processing alone. The sixth contrast, “DA vs both DA and pre-processing”, evaluates whether DA gives a lower error rate when used alone or when used together with pre-processing.

The seventh contrast, “Effect of DA with vs without pre-processing”, evaluates whether the effect of DA (i.e., the difference between the level for DA and the level for EM) is greater when used along with pre-processing than when used with one of the other assumed covariance structures. Positive values indicate that DA reduces the error rate more when pre-processing is not applied. The eighth contrast, “Effect of DA for unequal vs equal covariances”, evaluates whether the effect of DA is greater when the unconstrained covariance structure is assumed than when the equal but otherwise unconstrained covariance structure is assumed. Positive values indicate that DA reduces the error rate more when the groups’ covariance matrices are only assumed to be equal than when no covariance assumptions are made. The ninth contrast, “Effect of DA for equal vs equal and spherical”, evaluates whether the effect of DA is greater when the groups’ covariance matrices are assumed equal but are otherwise unconstrained than when the covariance matrices are assumed to all equal the identity matrix. Positive values indicate that DA reduces the error rate more when the groups’ covariance matrices are assumed to be identical and proportional to the identity matrix than they are only assumed to be identical. Finally, the tenth contrast, “Effect of pre-processing for DA vs EM”, evaluates whether the effect of pre-processing is greater when DA is used than when the EM algorithm is used. Positive values indicate that applying pre-processing improves the error rate more when the EM algorithm is used than when DA is used.

Our first observation about these results is that DA has a lower misclassification rate on average than the EM algorithm. This effect is consistent across assumptions about the mixture proportion and covariance structure, as well as across interactions with many of the data generation variables. The effect is much smaller however when DA is used with pre-processing.

Not assuming that the mixture proportions are equal gives a slightly (but significantly) smaller error rate than when we make this assumption. One particularly interesting interaction for this factor is with the true mixture proportion, p_1 (See Figure 3.1). When $p_1 = 0.25$ (i.e., proportions are somewhat different), not assuming equal mixture proportions is preferable; when $p_1 = 0.5$ (proportions are the same), assuming equal mixtures is better; and when $p_1 = 0.35$ (proportions are only somewhat different) it makes little difference.

The ranking of assumed covariance structures in order from lowest error rate to highest is as follows: identical and proportional to the identity matrix with pre-processing, unconstrained, identical and proportional to the identity matrix without pre-processing, and identical but otherwise unconstrained. If we consider the interaction between assumed covariance structure and method, however (see Figure 3.5), we see that the error rate when using unconstrained covariances is now lower than the one using pre-processing. Further, when combined with DA, pre-processing gives the highest error rate, followed by identical and proportional to the identity matrix without pre-processing and identical but otherwise unconstrained. The ordering of covariance structures remains unchanged when the EM algorithm is used.

The combinations of decision variables with the lowest average error rate is DA with completely unconstrained covariances, and with or without assuming that the mixture proportions are equal.

3.2 k-Means Comparison

3.2.1 Overview and Parameter Settings

As in the previous simulation, we begin this section by describing our statistical analysis, and identifying which effects have a significant influence on the misclassification rate. We then present some interesting interaction plots and contrasts.

The purpose of this simulation is to compare DA with the EM and CEM algorithms when assuming the most simple model structure, equal mixture proportions with all covariance matrices equal and proportional to the identity matrix. This model structure is chosen because fitting it using the CEM algorithm is equivalent to the very popular k-means algorithm (Celeux and Govaert, 1992). We also investigate whether Yang's pre-processing step has any effect on the misclassification rate. Note that, as in the previous simulation, models that use pre-processing or DA (or both) are considered supervised while all others are considered unsupervised.

The details of this simulation are nearly identical to the one described above. A split-plot design is employed, where the whole-plot factors are the data generation variables: mean difference, mixture proportions, dimension, covariance structure and sample size. These factors all take the same levels as in the EM vs DA simulation except for sample size, which now takes the values $300, 600, 900$ and 1200. These larger sample sizes are used to more

Table 3.10: Factors for the k-means comparison simulation, with their numbers of levels and whether they are applied to data generation or data analysis.

Factor	Levels	Type
Mean Difference	5	Generation
Mixture Proportion	3	Generation
Dimension	3	Generation
Sample Size	3	Generation
Covariance Structure	3	Generation
Pre-Processing	2	Analysis
Method	3	Analysis

closely match those in a different simulation by Celeux and Govaert (1993). The analysis variables make up the sub-plot factors, but we only vary the method and whether Yang's pre-processing procedure is applied. In order to be consistent with the k-means algorithm, we only consider analyses where the mixture proportions are assumed equal, and the groups' covariance matrices are all assumed equal and proportional to the identity matrix. Yang's pre-processing procedure is either applied or not, and the methods considered are DA, the EM algorithm and the CEM algorithm. The EM and CEM algorithms both require multiple re-starts (as described above), so we run each 20 or 45 times (depending on the difference between the means) and choose the best result. The factors we consider, along with their position within the split-plot structure, are listed in Table 3.10.

Once a dataset and analysis method are selected, the model is fit and 1000 new observations are used to estimate the misclassification rate in same way as in the previous simulation[4].

We generate our data using `R` (R Core Team, 2016) and analyze them using `SAS`® software (SAS Institute Inc., 2013) in the same way as the previous simulation. All the same functions are used for data generation and analysis, but here we also use the `kmeans` function in base `R` to implement the k-means algorithm.

We observe the phenomenon discussed above here as well, where a cluster consists of a single observation. This leads to the same computational problem with estimating a covariance matrix for that cluster, and we address this problem in the same way. That is, whenever this occurs for a single model, we generate a new dataset (with the same dataset variables) and re-fit all the models. This phenomenon is even more rare here, and occurs in less than 0.04% of cases (5 of 16200 total model fits). All these instances occur when Yang's method is used with DA, but it is difficult to infer anything from this due to the limited number of cases.

[4] And again, the difficulty of assigning group labels to clusters is avoided by labeling clusters in the best possible way.

Table 3.11: Data generation effects in the k-means comparison simulation that are significant at the $\alpha = 0.05$ level. Effects that are are also significant at the Bonferroni corrected level are indicated with an exclamation point (see text).

Source	
Mean Difference	!
Mixture Proportion	!
Mean Difference*Mixture Proportion	!
Dimension	
Mean Difference*Dimension	!
True Covariance Structure	!
Mean Difference*True Covariance Structure	!
Mixture Proportion*True Covariance Structure	!
Mean Difference*Mixture Proportion*True Covariance Structure	
Dimension*True Covariance Structure	!
Mean Difference*Dimension*True Covariance Structure	!
Sample Size	

3.2.2 Analysis

The analysis of this simulation parallels that of our EM vs DA simulation. We analyze our data as a split-plot design, where the data generation variables correspond to the whole-plots and the analysis variables correspond to sub-plots. Our response variable is the misclassification rate of a particular analysis on a single dataset. As above, we apply the appropriate variance stabilizing transformation for this response.

A mixed-effects linear model is again used, and interactions of order higher than 3 are excluded. Effects that are significant at the $\alpha = 0.05$ level are listed in Tables 3.11-3.13. Using the Bonferroni correction here suggests setting $\alpha = 7.9 \cdot 10^{-4}$. We indicate effects that are also significant at this level by adding an exclamation point. Table 3.11 contains effects that pertain only to data analysis, while Tables 3.12 and 3.13 list all effects that relate to whether pre-processing is used, and which fitting method is used, respectively. Note that some terms occur in both Tables 3.12 and 3.13 because they contain both of these variables. As mentioned above, we are particularly interested in studying the analysis variables so that we can make recommendations for analysts.

We now investigate the estimated misclassification rate for these effects. As above, we are primarily interested in the pre-processing and method effects, since levels of these are chosen by the analyst and we would like to make practical recommendations. A complete analysis would be too much space to include here, so we only present highlights.

Estimates of misclassification rate for each analysis factor are given in Tables 3.14 and 3.15. The differences between these factors' levels are all significant, even after applying Tukey's correction (Milliken and Johnson, 1992). Some two-way interaction plots are given in Figures 3.6-3.9. Note that the covariance structures "Hetero", "Homo" and "Sph" cor-

Table 3.12: Effects in the k-means comparison simulation containing “Pre-Processing”, that are significant at the $\alpha = 0.05$ level. Effects that are are also significant at the Bonferroni corrected level are indicated with an exclamation point (see text).

Source	
Pre-Processing	!
Pre-Processing*Method	!
Mean Difference*Pre-Processing	!
Mean Difference*Pre-Processing*Method	!
Mixture Proportion*Pre-Processing	!
Mixture Proportion*Pre-Processing*Method	!
Mean Difference*Mixture Proportion*Pre-Processing	!
Dimension*Pre-Processing	!
Dimension*Pre-Processing*Method	!
Mean Difference*Dimension*Pre-Processing	!
True Covariance Structure*Pre-Processing	!
True Covariance Structure*Pre-Processing*Method	!
Mean Difference*True Covariance Structure*Pre-Processing	!
Mixture Proportion*True Covariance Structure*Pre-Processing	!
Dimension*True Covariance Structure*Pre-Processing	!
Mixture Proportion*Sample Size*Pre-Processing	
True Covariance Structure*Sample Size*Pre-Processing	!

Table 3.13: Effects in the k-means comparison simulation containing “Method”, that are significant at the $\alpha = 0.05$ level. Effects that are are also significant at the Bonferroni corrected level are indicated with an exclamation point (see text).

Source	
Method	!
Pre-Processing*Method	!
Mean Difference*Method	!
Mean Difference*Pre-Processing*Method	!
Mixture Proportion*Method	!
Mixture Proportion*Pre-Processing*Method	!
Mean Difference*Mixture Proportion*Method	!
Dimension*Pre-Processing*Method	!
Mean Difference*Dimension*Method	!
Mixture Proportion*Dimension*Method	!
True Covariance Structure*Method	!
True Covariance Structure*Pre-Processing*Method	!
Mean Difference*True Covariance Structure*Method	!
Mixture Proportion*True Covariance Structure*Method	
Dimension*True Covariance Structure*Method	!
Sample Size*Method	

Table 3.14: Estimated error rate for each level of pre-processing in the k-means comparison simulation.

Pre-Processing	Estimate
No	0.1413
Yes	0.1259

Table 3.15: Estimated error rate for each level of method in the k-means comparison simulation.

Method	Estimate
CEM	0.1348
EM	0.1725
DA	0.0990

respond respectively to completely unconstrained covariance matrices, equal but otherwise unconstrained covariance matrices and all covariance matrices equal and proportional to the identity matrix (i.e., spherical groups). Relevant contrasts, along with their estimates, standard errors and p-values, are given in Table 3.16. As above, the estimates and standard errors of these contrasts are on the transformed scale and smaller values correspond to lower misclassification rates in the direction suggested by the contrast's description. We divide these contrasts into the same two groups discussed in the previous simulation. The first group consists of the first three contrasts, while the second group consists of the last two contrasts. We describe the meaning of these contrasts here, and discuss their significance in Section 4.1.

The first contrast, "DA without Pre-Processing", evaluates whether DA is better than the EM or CEM algorithms when Yang's pre-processing procedure is not used. The second contrast, "Pre-Processing with EM and CEM", evaluates whether Yang's pre-processing procedure is preferable when the fitting method is unsupervised (i.e., EM or CEM). The third contrast, "DA and pre-processing vs one without the other", evaluates whether using DA and Yang's pre-processing outperforms the average of Yang's pre-processing with the EM algorithm, and DA with the unsupervised covariance structures.

The fourth contrast, "Effect of pre-proc for EM vs CEM", evaluates whether the effect of pre-processing (i.e., the difference between the level for the pre-processed covariance structure and the average of the other assumed covariance structures) is greater when used with the EM algorithm than with the CEM algorithm. Positive values here indicate that pre-processing reduces the error rate more when the EM algorithm is used. Finally, the fifth contrast, "Effect of pre-proc for DA vs others", evaluates whether the effect of pre-processing is greater when used with DA than when used with the EM or CEM algorithms.

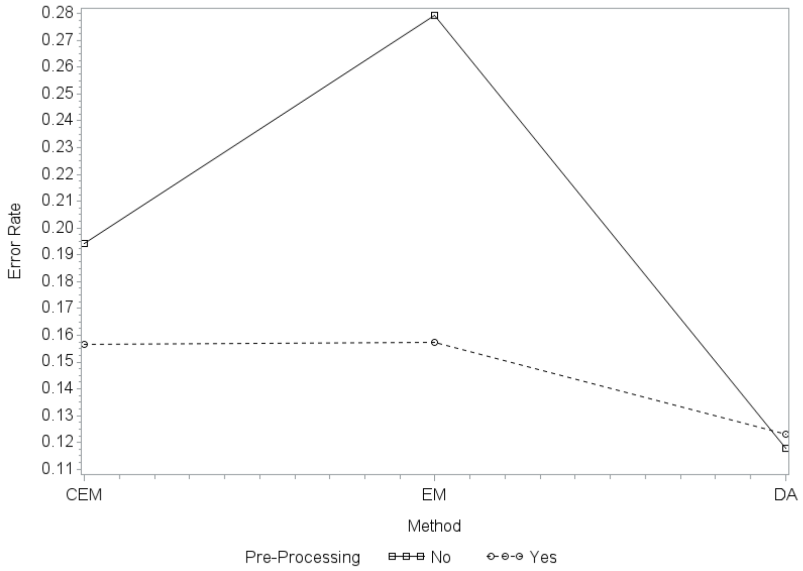


Figure 3.6: Plot of the interaction effect between fitting method and whether pre-processing is applied in the k-means comparison simulation.

Positive values here indicate that pre-processing reduces the error rate more when DA is used.

Table 3.16: Interesting contrasts for the EM vs DA simulation. See the text for a detailed description of the meaning of each contrast.

Label	Estimate	Standard Error	P-Value
DA without Pre-Processing	-0.1042	0.0012	<.0001
Pre-Processing with EM and CEM	-0.0413	0.0010	<.0001
Both Supervisions vs Single Supervision	-0.0080	0.0012	<.0001
Effect of pre-processing for EM vs CEM	0.0904	0.0020	<.0001
Effect of pre-processing for DA vs others	-0.0089	0.0009	<.0001

Our first observation here is that, unsurprisingly, analyses with supervision tend to outperform those without. That is, DA has lower average misclassification rate than both the EM and CEM algorithms, and error rates are lower on average when Yang's pre-processing step is applied than when it is not. Using DA with pre-processing is redundant however, and actually increases the error rate slightly compared to DA alone. This is consistent with the results in the previous simulation.

Among unsupervised methods, the CEM algorithm appears to outperform the EM algorithm, and this trend is consistent across numerous interactions. The difference becomes smaller in higher dimensions, but remains fairly constant across sample sizes.

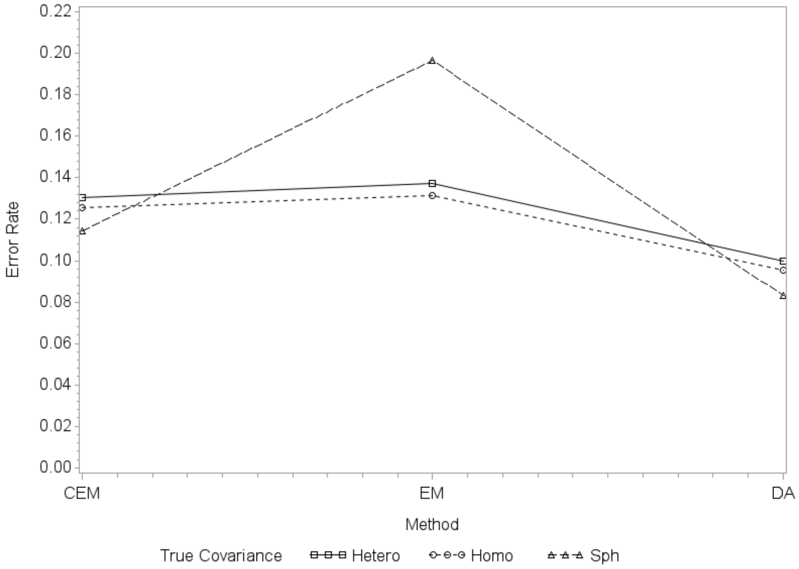


Figure 3.7: Plot of the interaction effect between fitting method and true covariance structure in the *k*-means comparison simulation.

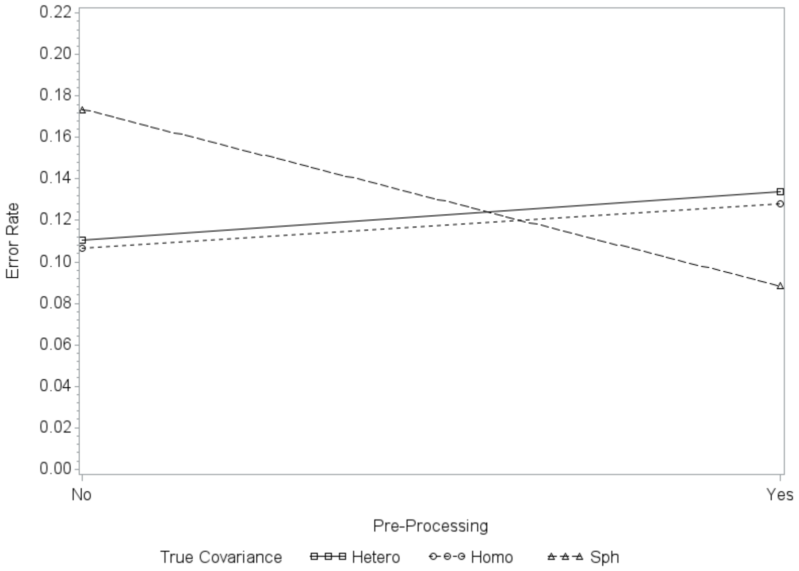


Figure 3.8: Plot of the interaction effect between whether pre-processing is applied and the true covariance structure in the *k*-means comparison simulation.

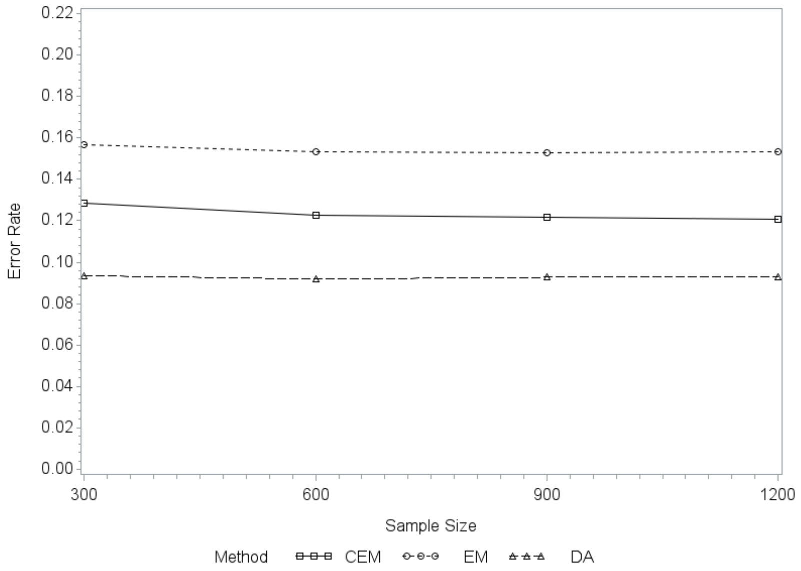


Figure 3.9: Plot of the interaction effect between sample size and method in the *k*-means comparison simulation.

The combination of decision variables that gives the lowest average error rate is DA without pre-processing.

Chapter 4

Discussion

4.1 Evaluation and Recommendations

The primary goal of this study is to investigate whether unsupervised clustering methods are appropriate to use for solving discrimination problems. Our simulation studies suggest that the answer to this question is no. In the EM vs DA simulation, incorporating information about the response through either the fitting method (DA) or assumed covariance structure (using pre-processing when all groups' matrices are equal and proportional to the identity matrix) is preferable, on average, to the other unsupervised options. We see a similar trend in the k-means comparison simulation where, even after introducing another unsupervised method for DA to compete against, it still outperforms the unsupervised methods.

A secondary goal of this study is to investigate the merit of Yang's pre-processing procedure compared to other model-based clustering and discrimination methods. As we just discussed, supervision appears to be important for discrimination, and Yang's procedure does add supervision to a clustering problem that may be otherwise unsupervised. However, DA appears to outperform Yang's procedure when averaged across all datasets. Further, using DA alone gives a lower error rate than either DA with pre-processing or pre-processing by itself. We therefore recommend using DA in place of Yang's pre-processing procedure when analyzing data.

We would also like to make recommendations about assumed covariance matrices. Recall that, in the EM vs DA simulation, the making no assumptions about the groups' covariance matrices is one of the best choices for both methods. In particular, the lowest error rate is obtained by using DA with no covariance assumptions. This suggests that the sample sizes we use are sufficiently large, on average, to estimate multiple unconstrained covariance matrices. This is comforting, as the average sample size in this simulation is 400, which is not too large to be conceivably obtained in many settings. Note however, that we only investigate low-dimensional problems, and that much larger samples may be required as the dimension increases.

The last decision variable that we investigate is whether or not to assume equal mixture proportions. Unsurprisingly, this is tied to the true values of the mixture proportions. If the proportions are somewhat disparate (e.g., 0.25 and 0.75) then the error rate is higher when we assume that they are equal. Inversely, if the mixture proportions are equal, the error rate goes down when we assume that they are equal. If the proportions are only slightly disparate (e.g., 0.35 and 0.65) then it makes little difference what we assume. It is therefore important to use any a priori knowledge about a problem to select the appropriate assumption. Lacking any knowledge whatsoever, an uninformative prior (Strachan and van Dijk, 2003) on the mixture proportions suggests that they will be disparate more often than similar, so we should not assume that they are equal.

We conclude with recommendations for data analysts using these models. Considering both simulations, we recommend using DA with no constraints on the covariance matrices, and mixture proportions assumed equal. Our simulations do not cover all possible models (e.g., the CEM algorithm is only applied for one covariance structure, with and without pre-processing), but DA is the best method in both simulations, and the k-means comparison simulation does not give any indication that the CEM algorithm can outperform DA in any of the settings that we do not consider. This supports the popularity of QDA as a supervised learning method, and suggests that it may be appropriate in low-dimensional problems with samples as small as 200.

4.2 Limitations and Possible Extensions

Many limitations of this study are computational in nature. The `Rmixmod` package (Lebret et al., 2015) in `R` is an interface to the `MIXMOD` software package (Biernacki et al., 2006) that allows for models with any of the 14 covariance structures listed in Table 2.1 to be fit using DA or the EM or CEM algorithms. This package also allows us to optionally assume equal mixture proportions. Unfortunately, it provides limited feedback when models cannot be fit correctly. This makes implementing large-scale simulations using this software impossible. Other implementations of model-based clustering in `R` are more user friendly, but do not have the same scope as the `MIXMOD` package. This is why our simulations do not consider all combinations of the data analysis factors. It would be interesting in future work to investigate how models with other covariance structures fit using the CEM algorithm compare with DA and the EM algorithm, but this comparison may require writing a new implementation. Doing so would have been outside the scope of this project.

Another computational limitation arises from our proposed solution for handling the case where an extreme outlier is assigned to its own cluster by the fitting procedure. Our solution is to draw a new dataset whenever this phenomenon is observed, and re-fit all models at that iteration to the new dataset. This creates a new problem however, because the data analyzed when this solution is employed no longer follow a mixture normal distribution, but instead follow a similar distribution from which observations are more likely to be tightly packed. This is because any sample with an outlier that is sufficiently extreme to cause the above phenomenon is assigned probability zero, thereby slightly increasing the probabilities of all other possible samples. It is not clear to the author how to satisfactorily address this problem. Fortunately, it occurs very rarely across both simulations, but analysts should be most cautious when using the EM algorithm or allowing the mixture proportions to be unequal.

A third computational limitation is related to time. In some cases, it would be of interest to investigate higher-order interaction terms between factors in our simulations. Unfortunately, fitting a mixed model of this size with even third-order interactions takes a long time, and the trade-off between learning somewhat more about a small number of effects versus growing computation time exponentially seems too costly.

A more conceptual limitation is that we only consider two groups and dimensions up to six. The procedures discussed in Chapter 2 can all be used to fit models with any number of groups in any dimensions. We focus on the limited case of two groups in order to develop an understanding of how the methods behave in this simple case. As discussed previously, we limit the number of dimensions to ensure that all models can be fit using the `R` package we employ. Future work may consist of investigating whether our results continue to hold when the number of groups is larger than two or when the dimension is larger than six.

Another conceptual limitation is that we only consider data from mixture-normal distributions, and all of our methods assume data of this form. That is, we do not consider these methods' robustness to the assumption of normally distributed data. It would be interesting in future research to include this as another factor in simulation studies. In particular, the Laplace distribution, which is commonly used to simulate data with more outliers (i.e., heavier tails) than the normal (Kotz et al., 2001), would be a good place to start.

In order to ensure uniform sizes across groups, we constrain all covariance matrices in our simulations to have determinant 1. This is a limitation however, since we are unable to estimate the influence of different sizes of covariance matrices on the models we consider. Specifically, allowing covariance matrices to have different determinants violates the constraints imposed in three of our four covariance structures (all but completely unconstrained). Future investigation may be able to clarify this topic.

Many clustering algorithms exist that are not model-based (or at least for which no corresponding model has been found). Examples of these include the hierarchical clustering algorithms known as single-linkage and complete-linkage (Fraley and Raftery, 1998; Hastie et al., 2011). It would be interesting to identify under which circumstances we can expect these algorithms to outperform the model-based methods we study, and whether supervision changes the performance of these non-model-based methods.

A strange phenomenon occurs in Figure 3.4, specifically when the true covariance structure is "homoscedastic" or "spherical". We see here that, although the correct assumption is that the groups' covariance matrices are all equal, or all equal and proportional to the identity matrix respectively, the assumed covariance structure with the lowest misclassification rate is unconstrained (a weaker assumption than either homoscedastic or spherical). This is surprising because, due to the bias-variance trade-off (Hastie et al., 2011), we actually expect a more flexible assumed covariance structure to have a higher misclassification rate than a more restrictive but correctly specified structure. This is because assuming a more general structure means that more parameters must be estimated than are necessary to describe the true structure, and estimating extra parameters adds variability to the resulting predictions. Having a sufficiently large sample would effectively negate this bias-variance trade-off by reducing the variance to negligible levels, but this does not explain reducing the error rate below the level obtained by assuming the correct structure. Further study is required to determine whether this is in fact a new phenomenon, or an error in our simulations.

Another curiosity is that there are several instances where the EM algorithm gives an unexpectedly high error rate (see e.g., Figures 3.5, 3.6 and 3.7). A similar phenomenon occurs when the true covariance structure is identical and equal to the identity matrix (see e.g., Figures 3.4 and 3.7). Figure 3.4 is particularly concerning, since the highest error rate for data with this covariance structure is obtained by making the correct assumption when fitting models. This is quite strange, and, as above, further study is required to identify whether this is a new phenomenon or an error in our simulations.

It is also interesting that in Table 3.15, the CEM algorithm outperforms the EM algorithm on average. This effect holds regardless of the sample size (See Figure 3.9). These findings contradicts the results in Celeux and Govaert (1993), who found that the CEM algorithm has a smaller error rate than the EM algorithm for small samples, and the reverse is true for large samples. This discrepancy may be due to the different parameter settings we consider, as Celeux and Govaert only consider data from groups with covariance matrices that are identical but otherwise unconstrained. However, we only investigate the CEM algorithm on data from groups with covariance matrices proportional to the identity matrix.

The Mahalanobis distance that we use in simulations is not exactly appropriate. Specifically, when groups' covariance matrices are completely unconstrained, we set the value of the Mahalanobis distance between the groups' means, relative to the covariance matrix of the second group. This does not take into account the distinct covariance matrix of the first group. It is not clear what the appropriate way to resolve this issue is. One possible solution is to set the Mahalanobis distance relative to the mean of the two covariance matrices (i.e., use $\tilde{\Sigma} = (\Sigma_1 + \Sigma_2)/2$), but there is no formal justification for this.

As discussed in Section 2.1.1, Yang proposes the $-\log(\text{p-value})$ transformation for his method with little justification. Specifically, he cites that this is a decreasing function of the p-value (therefore an increasing function of the evidence against the null hypothesis of no relationship between the predictor and response) and that it takes moderate values on the range that non-significant p-values usually take. It is possible therefore, that some other decreasing transformation may outperform the one he proposes. Examples include $1 - p$, $1/p$ and $1/\sqrt{p}$. A simulation similar to the one we perform could be used to investigate other transformations for pre-processing.

We study Yang's pre-processing procedure (Yang, 2013) as a variable weighting tool, but he also proposes a slight modification that allows it to be used for variable selection. By applying a threshold, and removing any variables with a Wald test p-value above this threshold, we can remove any variables that do not appear relevant for discrimination. Because it then functions as a screening procedure rather than a linear transformation, it could be applied to models that are invariant to such transformations. It would be interesting to see how this and other, more established variable selection techniques affect our clustering and discrimination methods.

Bibliography

S. Aerts, G. Haesbroeck, and C. Ruwet. Multivariate coefficients of variation: comparison and influence functions. *Journal of Multivariate Analysis*, 142:183–198, 2015.

Jeffrey D. Banfield and Adrian E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803–821, 1993.

Halima Bensmail and Gilles Celeux. Regularized gaussian discriminant analysis through eigenvalue decomposition. *Journal of the Americal Statistical Association*, 91(436):1743–1748, 1996.

C. Biernacki, G. Celeux, G. Govaert, and F. Langrognet. Model-based cluster and distcriminant analysis with the MIXMOD software. *Computational Statistics and Data Analysis*, 51(2):587–600, 2006.

Christopher R. Bilder and Thomas M. Loughin. *Analysis of Categorical Data with R*. Taylor and Francis Group, Boca Raton, FL, 2015.

P.A. Bromiley and N.A. Thacker. The effects of an arcsin square root transform on a binomial distributed quanity. Technical report, Medical School, University of Manchester, 2002.

Peter Bryant and John A. Williamson. Asymptotic behaviour of classification maximum likelihood estimates. *Biometrika*, 65(2):273–281, 1978.

George Casella and Roger L. Berger. *Statistical inference.* Duxbury, Pacific Grove, CA, 2002.

Vittorio Castelli and Thomas M. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16:105–111, 1995.

Vittorio Castelli and Thomas M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102–2117, 1996.

Gilles Celeux and Gérard Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14(3):315–332, 1992.

Gilles Celeux and Gérard Govaert. Comparison of the mixture and classification maximum likelihood in cluster analysis. *Journal of Statistical Computation and Simulation*, 43(3-4): 127–146, 1993.

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

Keith Conrad. Isometries of $\mathbb{R}^n$. `http://www.math.uconn.edu/~kconrad/blurbs/grouptheory/isometryRn.pdf`, 2016. [Online; accessed 23-May-2016].

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

Rick Durrett. *Probability: Theory and Examples.* Cambridge University Press, fourth edition, 2013.

Christoph F. Eick, Nidal Zeidat, and Zhenghong Zhao. Supervised clustering – algorithms and benefits. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 774–776, 2004.

David Firth. Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38, 1993.

Bernhard W. Flury, Martin J. Schmid, and A. Narayanan. Error rates in quadratic discrimination with constraints on the covariance matrices. *Journal of Classification*, 11: 101–120, 1994.

C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

Chris Fraley, Adrian E. Raftery, Thomas Brendan Murphy, and Luca Scrucca. *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*, 2012.

Stephen H. Friedberg, Arnold J. Insel, and Lawrence E. Spence. *Linear algebra.* Prentice Hall, Englewood Cliffs, NJ, 2 edition, 1989.

Bernard Haasdonk and Elżbieta Pękalska. Classification with kernel Mahalanobis distance classifiers. In Andreas Fink, Berthold Lausen, Wilfried Seidel, and Alfred Ultsch, editors, *Advances in Data Analysis, Data Handling and Busines Intelligence*, pages 351–361. Springer-Verlag, Berlin, Germany, 2009.

D. J. Hand. *Discrimination and classification.* John Wiley and Sons, 1981.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning.* Springer, New York, NY, 2011.

Georg Heinze and Michael Schemper. A solution to the problem of separation in logistic regression. *Statistics in Medicine*, 21:2409–2419, 2002.

Georg Heinze, Meinhard Ploner, Daniela Dunkler, and Harry Southworth. *logistf: Firth's bias reduced logistic regression*, 2013. URL `https://CRAN.R-project.org/package=logistf`. R package version 1.21.

Anil K. Jain. Data clustering: 50 years beyond *k*-means. *Pattern Recognition Letters*, 31: 651–666, 2010.

Samuel Kotz, Tomasz J. Kozubowski, and Krzysztof Podgórski. *The laplace distribution and generalizations.* Birkh'auser, 2001.

Robert O. Kuehl. *Design of Experiments: Statistical Principles of research design and analysis.* Brooks/Cole, Belmont, CA, 2000.

Rémi Lebret, Serge Iovleff, Florent Langrognet, Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Rmixmod: The R package of the model-based unsupervised, supervised, and semi-supervised classification Mixmod library. *Journal of Statistical Software*, 67(6): 1–29, 2015.

Ramon C. Littell, George A. Milliken, Walter W. Stroup, Russell D. Wolfinger, and Oliver Schabenberger. *SAS® for Mixed Models.* SAS Institute Inc., Cary, NC, second edition, 2006.

Richa Loohach and Kanwal Garg. Effect of distance functions on *k*-means clustering algorithm. *International Journal of Computer Applications*, 49(6):7–9, 2012.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In Lucien M. Le Cam and Jerzy Neyman, editors, *Proceedings of the Berkley symposium on mathematical statistics and probability*, Berkley, CA, 1967. University of California Press.

F. H. C. Marriott. 389: Separating mixtures of normal distributions. *Biometrics*, 31(3): 767–769, 1975.

Geoffrey McLachlan and David Peel. *Finite mixture models.* John Wiley and Sons, New York, NY, 2000.

Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions.* John Wiley and Sons, New York, NY, 2008.

Igor Melnykov and Volodymyr Melnykov. On k-means algorithm with the use of Mahalanobis distances. *Statistics and Probability Letters*, 84:88–95, 2014.

George A. Milliken and Dallas E. Johnson. *Analysis of messy data volume I: designed experiments.* Chapman and Hall, New York, NY, 1992.

Terence J. O'Neill. Normal discrimination with unclassified observations. *Journal of the American Statistical Association*, 73(364):821–826, 1978.

Bo Peng. The determinant: a means to calculate volume. `https://www.math.uchicago.edu/~may/VIGRE/VIGRE2007/REUPapers/FINALAPP/Peng.pdf`, 2007. [Online; accessed 18-July-2016].

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `https://www.R-project.org/`.

Revolution Analytics and Steve Weston. *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*, 2015. URL `https://CRAN.R-project.org/package=doParallel`. R package version 1.0.10.

SAS Institute Inc. *SAS/AF® 9.4 Procedure Guide.* Cary, NC, second edition, 2013.

A. J. Scott and M. J. Symons. Clustering methods based on likelihood ratio criteria. *Biometrics*, 27(2):387–397, 1971.

Rodney W. Strachan and Herman K. van Dijk. Bayesian model selection with an uninformative prior. *Oxford Bulletin of Economics and Statisticsl*, 65:863–876, 2003.

M. J. Symons. Clustering criteria and multivariate normal mixtures. *Biometrics*, 37(1): 35–43, 1981.

Larry Wasserman. *All of statistics.* Springer Science and Business Media, Inc., New York, NY, 2004.

Hadley Wickham. The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, 40(1):1–29, 2011. URL `http://www.jstatsoft.org/v40/i01/`.

Michael P. Windham. Parameter modification for clustering criteria. *Journal of Classification*, 4(2):191–214, 1987.

C. F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

Yuanyu Yang. Classification based on supervised clustering with application to juvenile idiopathic arthritis. Master's thesis, Simon Fraser University, 2013.