Mixture Codebook Classification
Part 2: Model Selection and Method Evaluation

Mette Langaas
Department of Mathematical Sciences
The Norwegian Institute of Technology
The University of Trondheim
mette@imf.unit.no

Abstract

This report is the companion to the report "Mixture Codebook Classification, Part 1: Method Outline" by Langaas (1995b).

A *mixture codebook* is a set of parameters and weights of a hierarchy of class-conditional finite mixture models. The concept of mixture codebook classification (MC^2) consists in estimating a mixture codebook and using it to construct a classification rule.

In this report aspects of model selection and parameter estimation are discussed. Model selection is concentrated in two areas; choice of component distributions, where Gaussian and multivariate Student-t distributions are considered, and choice of component architecture. Parameter estimation is concentrated on comparing different optimization criteria and on regularization. A small comparative study involving mixture codebook classification, static neural networks and statistical methods for discrimination and classification is presented.

Some keywords: Finite Mixture Models; Classification; Discriminant Analysis; Penalization; Regularization; Model Selection; EM Algorithm; Student-t Distribution; Conditional Likelihood.

Contents

1 Introduction

The problem of classification is regarded to be one of the most important statistical problems today and has applications in a wide range of fields. In classification an object is assumed to be a member of one of M predefined groups. The particular group is unknown and the goal is to assign the object to the correct group.

Classification is concentrated in two branches of techniques, those that model the class-conditional distributions, such as kernel discriminant analysis and linear discriminant analysis, and those that model the posterior probabilities directly, such as classification trees and multilayer perceptrons. The concept of mixture codebook classification (MC^2), Langaas (1995b), combines these two branches to produce a flexible tool for classification.

The term *mixture codebook* is an extension of the term codebook, used in the fields of neural networks and vector quantization, to include the parameters and weights of a hierarchy of class-conditional finite mixture models. Mixture models provide a useful alternative to nonparametric modelling, while being less restrictive than the usual distributional assumptions. The concept of mixture codebook classification (MC^2) consists in estimating a mixture codebook and using it to construct a classification rule.

A short outline of MC^2 and its relations to other methods are presented in section 2. Aspects of model selection and parameter estimation are then discussed. In section 3 model selection is concentrated in two areas; choice of component distributions and choice of component architecture. Gaussian and multivariate Student-t distributions are considered. Concerning parameter estimation, emphasis is put on comparing different optimization criteria and on regularization, in section 4. A small comparative empirical study involving mixture codebook classification, static neural networks and statistical methods for discrimination and classification is also presented. Conclusions and final comments are found in section 5.

2 Mixture Codebook Classification, The Total Setting

2.1 MC^2 Outline

This is a schematic outline of Mixture Codebook Classification (MC^2). For further details see Langaas (1995b) and the subsequent sections.

Problem In the problem of classification the aim is to predict the true class membership of an observation (object), $\boldsymbol{x}$. In the problem of discrimination the aim is to separate the sample space into disjoint regions for the classes.

Suppose that we have M predefined classes and let the ith class be denoted g_i, $i = 1, ..., M$. The classification of an observation, $\boldsymbol{x}$, can be done to the class with the maximum posterior class probability. We will call this MAP classification,

$$\operatorname*{argmax}_{i=1,\ldots,M} p(g_i|\boldsymbol{x}) = \operatorname*{argmax}_{i=1,\ldots,M} \frac{p(g_i)p(\boldsymbol{x}|g_i)}{\sum_{j=1}^{M} p(g_j)p(\boldsymbol{x}|g_j)}.$$

Here $p(g_i|\boldsymbol{x})$ denotes the posterior probability for class g_i, $p(g_i)$ is the prior probability for class g_i and $p(\boldsymbol{x}|g_i)$ is the class-conditional distribution for the observation $\boldsymbol{x}$

given the class g_i. The construction of the classification rule is based on a training set $\mathcal{T} = (\boldsymbol{x}_j, \boldsymbol{z}_j)_{j=1}^n$ of n independent observations and known encoded class memberships. The encoding $\boldsymbol{z}_j$ has elements $\{z_{ji}\}$ where $z_{ji} = 1$ if the jth observation belongs to class g_i and $z_{ji} = 0$ otherwise.

MC^2 Models The MC^2 is built on modelling the class-conditional distributions as finite mixture models, $p(\boldsymbol{x}|g_i) = p(\boldsymbol{x}|g_i, \boldsymbol{\Theta})$. Two types of models are used, specific component densities for each class and common component densities for all classes.

Common Components Let the lth component be denoted c_l and the lth component density be denoted f_l, $l = 1, ..., K$. The following parametric model is provided for the class-conditional distributions,

$$p(\boldsymbol{x}|g_i, \boldsymbol{\Theta}) = \sum_{l=1}^{K} p(c_l|g_i) f_l(\boldsymbol{x}|\boldsymbol{\theta}_l). \tag{1}$$

This is a finite mixture model with mixing weights, $p(c_l|g_i)$, that are non-negative and sum to one, $\sum_{l=1}^{K} p(c_l|g_i) = 1$, and component densities, $f_l(\boldsymbol{x}|\theta_l)$.
A *mixture codebook*, $\mathcal{M} = \{\boldsymbol{\Theta}, f, K\}$, consists of the parameters and weights, $\boldsymbol{\Theta} = \{(\boldsymbol{\theta}_l, p(c_l|g_i), p(g_i)); i = 1, ..., M; l = 1, ..., K\}$, the component densities, $f = \{(f_l); l = 1, ..., K\}$ and the total number of components, K.

Class Specific Components Let the lth component and component density within the ith class, g_i, be denoted c_{il} and f_{il}, respectively, $l = 1, ..., k_i$. The following parametric model is provided for the class-conditional distributions,

$$p(\boldsymbol{x}|g_i, \boldsymbol{\Theta}) = \sum_{l=1}^{k_i} p(c_{il}|g_i) f_{il}(\boldsymbol{x}|\boldsymbol{\theta}_{il}). \tag{2}$$

Again, we have a finite mixture model with mixing weights, $p(c_{il}|g_i)$, that are non-negative and sum to one, $\sum_{l=1}^{k_i} p(c_{il}|g_i) = 1$ for $i = 1, ..., M$ and component densities, $f_{il}(\boldsymbol{x}|\theta_{il})$, that are specific for each component within each class.
The *mixture codebook*, $\mathcal{M} = \{\boldsymbol{\Theta}, f, K\}$, consists of the parameters and weights $\boldsymbol{\Theta} = \{(\boldsymbol{\theta}_{il}, p(c_{il}|g_i), p(g_i)); i = 1, ..., M; l = 1, ..., k_i\}$, the component densities, $f = \{(f_{il}); i = 1, ..., M; l = 1, ..., k_i\}$ and the vector with the number of components for each class, $K = (k_1, k_2, ..., k_M)$.

In both these cases, common components and class specific components, K is seen as giving the component architecture. The MC^2 model with common components is given a graphical presentation in figure 1 and with class specific components in figure 2. We see that the model with class specific components can be found as a special case of common components where some mixing weights are equal to zero.

The solution to the problem of classification is completely specified by the mixture codebook. That is, given the mixture codebook the classification rule is known. The problem of classification is transformed into the problem of choosing and estimating the mixture codebook.

The issues of choosing the component distributions and the component architecture are investigated in section 3 on model selection.

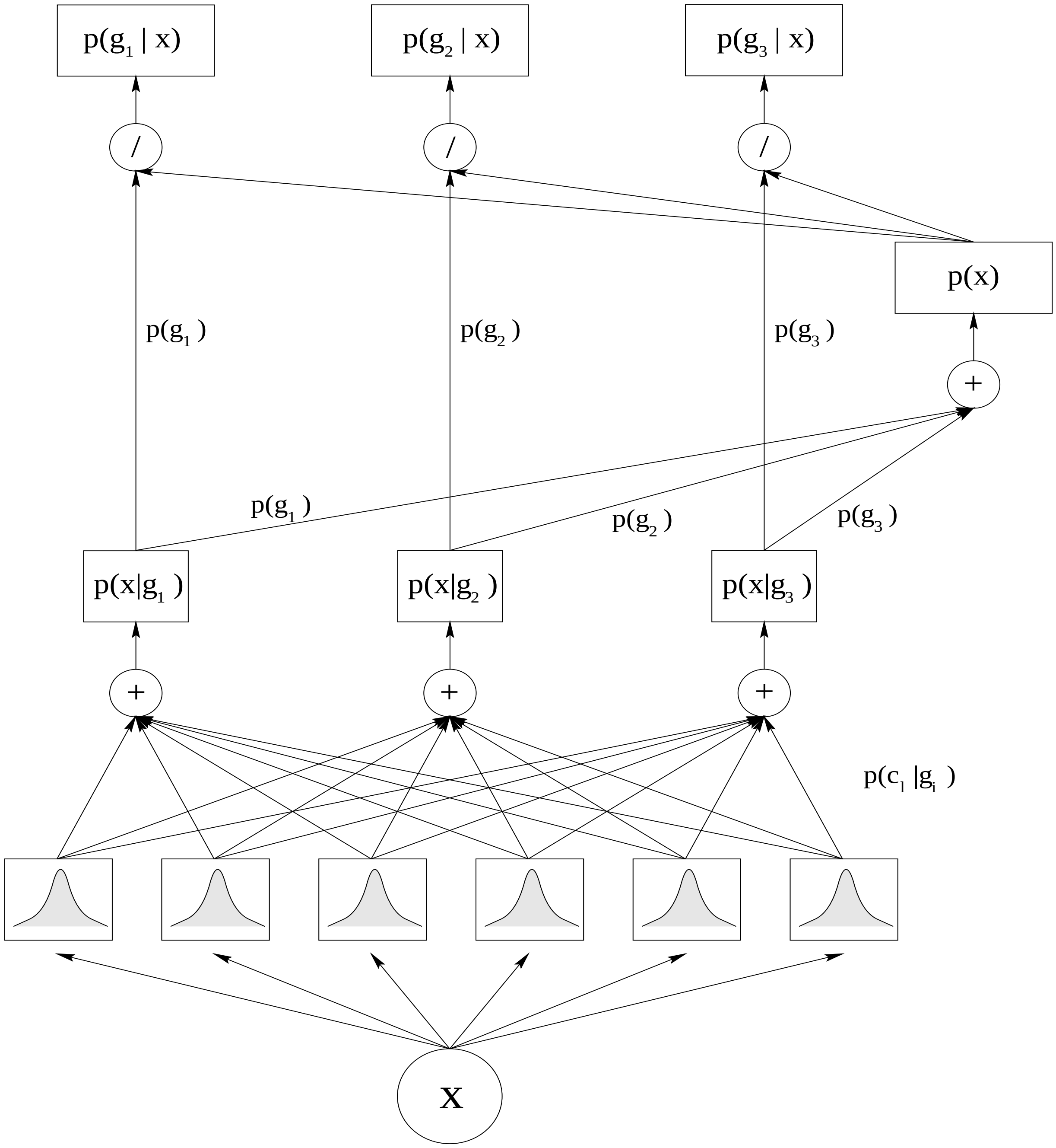


Figure 1: MC^2 model with common components for the classes, component architecture $K = 6$.

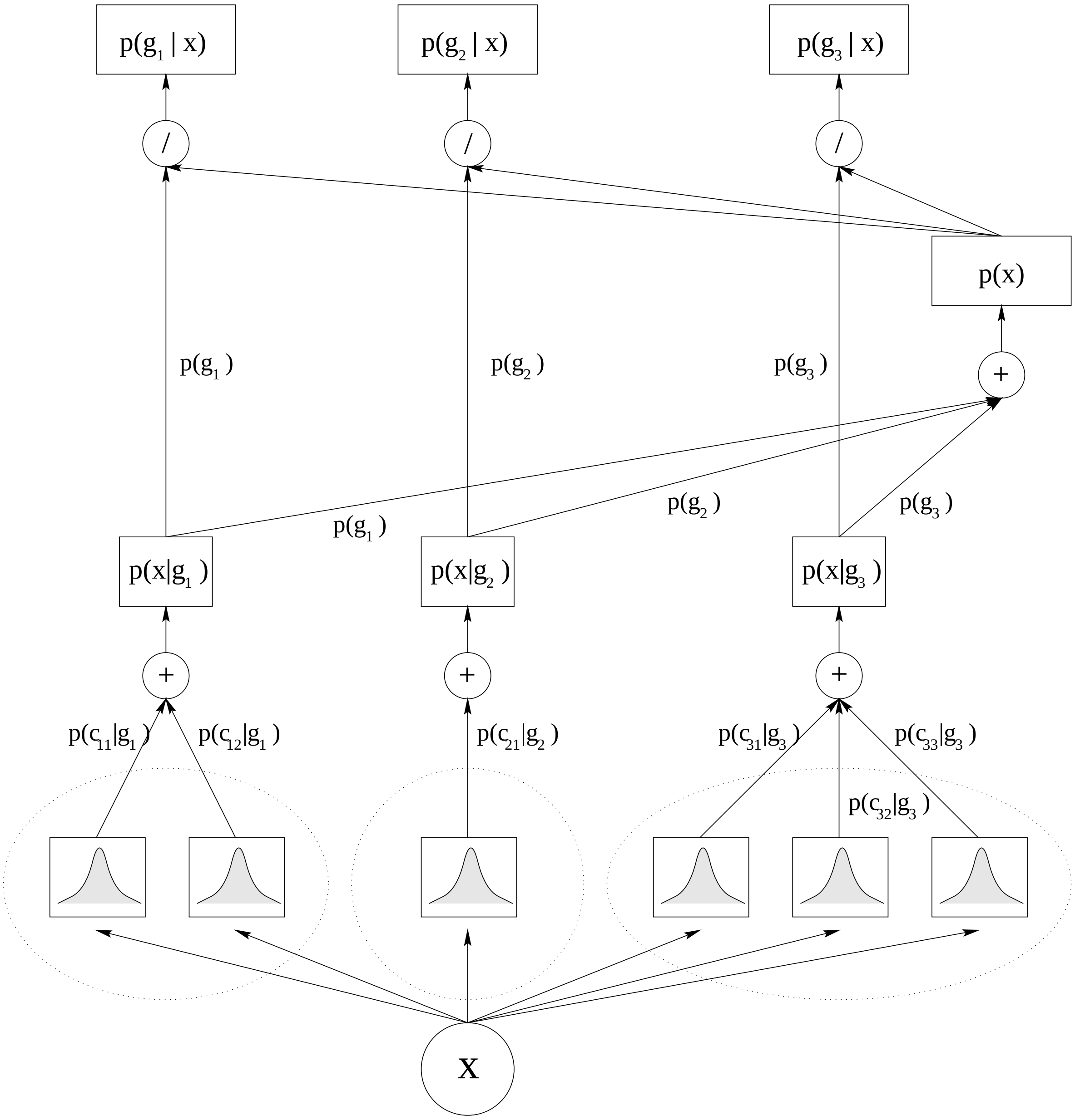


Figure 2: MC^2 model with class specific components, component architecture $K = (2, 1, 3)$.

MC^2 Methods Given the choice of model, choice of component distribution (f) and choice of component architecture (K), the task is to estimate the parameters and weights, $\mathbf{\Theta}$, on the basis of the observations from the training set, $\mathcal{T}$. That is, given $\{f, K\}$, the parameters and weights, $\mathbf{\Theta}$, are estimated from the training set with the aim of optimizing a criterion function. Different criterion functions, $\mathcal{J}(\mathbf{\Theta}|\mathcal{T})$, are studied.

(a) Maximize the joint likelihood (JOINT)

$$\mathcal{J}(\mathbf{\Theta}|\mathcal{T}) = \log(\prod_{j=1}^{n} p(\boldsymbol{x}_j, \boldsymbol{z}_j)) = \sum_{j=1}^{n}\sum_{i=1}^{M}(z_{ji}\log p(g_i) + z_{ji}\log p(\boldsymbol{x}_j|g_i, \mathbf{\Theta})) \tag{3}$$

(b) Maximize the conditional likelihood (COND)

$$\mathcal{J}(\mathbf{\Theta}|\mathcal{T}) = \log\prod_{j=1}^{n}\prod_{i=1}^{M}[p(g_i|\boldsymbol{x}_j)^{z_{ji}}] = \sum_{j=1}^{n}\sum_{i=1}^{M} z_{ji}\log p(g_i|\boldsymbol{x}_j, \mathbf{\Theta}) \tag{4}$$

(c) Minimize a quadratic loss function (QUAD) (least squares)

$$\mathcal{J}(\mathbf{\Theta}|\mathcal{T}) = \sum_{j=1}^{n}\sum_{i=1}^{M}(z_{ji} - p(g_i|\boldsymbol{x}_j, \mathbf{\Theta}))^2 \tag{5}$$

(d) Minimize the Kullback-Leibler divergence for independent Bernoulli variables (KL)

$$\mathcal{J}(\mathbf{\Theta}|\mathcal{T}) = \sum_{j=1}^{n}\sum_{i=1}^{M}(z_{ji}\log\frac{z_{ji}}{p(g_i|\boldsymbol{x}_j, \mathbf{\Theta})} + (1 - z_{ji})\log\frac{1 - z_{ji}}{1 - p(g_i|\boldsymbol{x}_j, \mathbf{\Theta})}) \tag{6}$$

(e) Minimize the misclassification rate (MISC)

$$\mathcal{J}(\mathbf{\Theta}|\mathcal{T}) = \sum_{j=1}^{n} I(\operatorname*{argmax}_{i=1,M} p(g_i|\boldsymbol{x}_j, \mathbf{\Theta}) \neq \operatorname*{argmax}_{i=1,M} \boldsymbol{z}_j) \tag{7}$$

where $I(A) = 1$ if A is true and $I(A) = 0$ otherwise.

(f) Combine two of the above criteria for a partition of the parameters and weights, $\mathbf{\Theta}$, into $\mathbf{\Theta} = (\mathbf{\Theta}_1, \mathbf{\Theta}_2)$. For example, estimate the parameters and weights, $\mathbf{\Theta}$, by maximizing the joint likelihood. Keep part of the estimates in $\mathbf{\Theta}_1$ and find new estimates of the remaining part, $\mathbf{\Theta}_2$, by maximizing the conditional likelihood. This will be regarded as a penalization problem and is discussed in section 4.2 on regularization.

The fitted model is denoted $p(g_i|\boldsymbol{x}, \hat{\mathbf{\Theta}})$, $i = 1, ..., M$.

All of the above criteria can also be regularized by including a penalty term or a condition on the solution. The criteria can also be modified by weighting the observations. This is of interest if the proportions in the training set do not reflect the underlying prior probabilities for the classes or if some misclassifications are more costly than others. For the misclassification criterion (7) classification to the class with the largest posterior probability can be substituted by other rules that are based on the posterior probabilities for the classes. Emphasis will in the following be on using the joint and the conditional likelihood criteria for classification to the class with the largest posterior probability (MAP classification).

Algorithms Different algorithms are implemented depending on the chosen optimization criterion.

1. The joint likelihood criterion (3) is maximized using the EM algorithm. The EM algorithm for Gaussian mixtures is outlined in section 3.4 and 3.5 of Langaas (1995b) and for Student-t mixtures in section 3.1.2 of this report.
2. For the other criterion functions a downhill simplex search routine is used, Press et al. (1988).

Regularization can also be done implicitly in the chosen algorithm by early stopping.

Solution A "soft" classification is supplied, that is, at a given observation, $\boldsymbol{x}$, the posterior probability for each class is given. The "hard" classification can be done using MAP classification or alternatively by choosing the class that minimizes some total cost of misclassification with the possibility of using different costs for each misclassification.

We will in this report look into the issues of model selection (choice of component distributions and component architecture) and parameter estimation and method evaluation (comparison of criteria functions and regularization).

2.2 MC^2 in Relation to Other Methods

Many methods for discrimination and classification are closely related to the MC^2 framework.

Linear and Quadratic Discriminant Analysis are special cases of the MC^2 framework, where the joint likelihood (3) is maximized using a MC^2 model with class-specific Gaussian component densities with one component inside each class. Linear discriminant analysis has equal covariance matrices for the classes and quadratic discriminant analysis has class specific covariance matrices. For details on linear and quadratic discriminant analysis see for example Johnson and Wichern (1982).

Logistic/Log-Linear Discrimination For the MC^2 framework with Gaussian class specific component densities with one component inside each class and equal covariance matrices for the classes maximizing the conditional likelihood will result in logistic (also called log-linear) discrimination, as described for example in Seber (1984).

Kernel Discriminant Analysis and Probabilistic Neural Networks In kernel discriminant analysis, Hand (1982) and McLachlan (1992), and in probabilistic neural networks (PNN), Specht (1990), the class-conditional distributions are estimated using a kernel density estimate,

$$\hat{p}(\boldsymbol{x}|g_i) = \frac{1}{n_i}\sum_{j=1}^{n} z_{ji}K(\boldsymbol{x}-\boldsymbol{x}_j) = \frac{1}{n_i}\frac{1}{\sigma^d}\sum_{j=1}^{n} z_{ji}K'(\frac{\boldsymbol{x}-\boldsymbol{x}_j}{\sigma})$$

where K and K' are kernel function, n_i is the number of observations from class g_i in the training set, $z_{ji}=1$ if $\boldsymbol{x}_j \in g_i$ and zero otherwise, σ is the bandwidth and d is the dimension of the observations. If the kernels are nonnegative the kernel density estimation can be interpreted as using equal weights mixtures with one mixture component centred at each observation, section 2.2 of Titterington et al. (1985). Classification is based on the posterior distributions for the classes. In PNN multivariate kernels are constructed as products of univariate kernels and the same amount of scaling (bandwidth) is used for all kernel functions.

The kth nearest neighbour method is a special case of the kernel discriminant analysis. If we estimate the prior probabilities of the classes by the proportions n_i/n of the observations in the training set and use adaptive kernel functions that are constant over the nearest k observations and zero elsewhere, we have the following expression for the posterior distribution of the classes,

$$\hat{p}(g_i|\boldsymbol{x}) = \frac{\frac{n_i}{n}\frac{1}{n_i}\sum_{j=1}^{n} z_{ji}K(\boldsymbol{x}-\boldsymbol{x}_j)}{\sum_{m=1}^{M}\frac{n_m}{n}\frac{1}{n_m}\sum_{j=1}^{n} z_{jm}K(\boldsymbol{x}-\boldsymbol{x}_j)} = \frac{\sum_{j=1}^{n} z_{ji}K(\boldsymbol{x}-\boldsymbol{x}_j)}{\sum_{j=1}^{n} K(\boldsymbol{x}-\boldsymbol{x}_j)}$$

Classification to the class with the largest posterior probability using this estimate for the posterior probabilities of the classes is the kth nearest neighbour method. The kth nearest neighbour method is described further in Fix and Hodges (1951) and Dasarathy (1991).

Learning Vector Quantizer of Kohonen (1990) creates a codebook from the training set and uses the codebook in place of the training set for first nearest codeword classification. This can be seen as modelling the class-conditional distributions by class specific component distributions that are uniform over the Dirichlet tessellation formed by the codewords within each class.

Mixture Discriminant Analysis is a recent contribution by Hastie and Tibshirani (1995). The MDA is based on modelling the class-conditional distributions as class specific mixtures of Gaussian component densities with equal covariance matrices for all components. Parameters are estimated by maximizing the joint likelihood (3) and classification is done to the class with the maximum posterior probability. MDA includes a feature that is called subclass shrinkage which regularizes the parameter estimation by penalizing the joint likelihood and using constraints on the rank of the matrix of the Gaussian component means. Hastie and Tibshirani (1995) have expressed the EM algorithm (found in Appendix A of Langaas (1995b)) as a repeated-regression procedure using optimal scoring and have thereby introduced the possibility of using nonparametric regression methods as described in Hastie et al. (1994) on flexible discriminant analysis (FDA). This means that the ability to produce nonlinear class boundaries is supplied both by the use of Gaussian mixtures and nonparametric regression method. Hastie and Tibshirani find that combining the standard MDA with FDA does not yield a much improved decision boundary, but they find that the two components might support each other in difficult situations. Another extension to the MDA includes the use of penalization as in Hastie et al. (1995) on penalized discriminant analysis (PDA). Hastie and Tibshirani find the PDA extension to MDA to be very promising and that it leads to variance reduction and increases the interpretability of the discriminant functions.

Gaussian Mixtures Classifiers is the name used by Lippmann (1994) for the method consisting in maximizing the joint likelihood for class-conditional class specific Gaussian mixtures, that is, the same basic idea as for MDA. The same idea has also been used by Taxt et al. (1991) and in the Gaussian clustering network of Tråvén (1991).

Radial Basis Function Networks The RBF networks are used both for regression and classification and are based on a wide variety of radial basis function type models and using different methods for parameter estimation.

Many types of RBF networks can be written in the following form, Xu et al. (1994),

$$a_i(\boldsymbol{x}) = \sum_{l=1}^{K} w_{il} h((\boldsymbol{x} - \boldsymbol{\mu}_l)^T \boldsymbol{\Sigma}_l^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_l))$$

where h is a basis function, $\boldsymbol{\mu}_l$ is the centre vector, w_{il} is a weight and $\boldsymbol{\Sigma}_l$ is a $d \times d$ positive scale matrix that controls the so called reseptive field of the basis functions. For simplicity let us write $r_l^2 = (\boldsymbol{x} - \boldsymbol{\mu}_l)^T \boldsymbol{\Sigma}_l^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_l)$. The most common choice of basis function is the Gaussian function, $h(r_l^2) = \exp(-r_l^2)$. Normalized RBF networks use normalized basis functions, $h'(\boldsymbol{u}) = h(\boldsymbol{u}) / \sum_{l=1}^{K} h(\boldsymbol{u})$, that is,

$$a_i(\boldsymbol{x}) = \frac{\sum_{l=1}^{K} w_{il} h(r_l^2)}{\sum_{l=1}^{K} h(r_l^2)}.$$

Denote the RBF parameters by $\boldsymbol{\Theta} = \{(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l, w_{il}); i = 1, ..., M; l = 1, ..., K\}$.

For classification the above RBF network can be seen as modelling the posterior probabilities of the classes, $a_i(\boldsymbol{x}) = p(g_i|\boldsymbol{x}, \boldsymbol{\Theta})$, as finite mixture models with component densities that are common to all classes (not class-conditional mixtures but posterior mixtures). A second interpretation is closely related to the MC^2 framework but is not commonly used. The RBF network can be seen as modelling the joint distribution for the observations and the classes, $a_i(\boldsymbol{x}) = p(g_i, \boldsymbol{x}|\boldsymbol{\Theta})$ so that $w_{il} = p(c_l, g_i) = p(c_l|g_i)p(g_i)$. This is equivalent to using class-conditional mixtures with common component densities for the classes as in (1). A third interpretation is based on the normalized RBF network. The normalized RBF network can be seen as modelling the posterior probabilities of the classes, $a_i(\boldsymbol{x}) = p(g_i|\boldsymbol{x}, \boldsymbol{\Theta})$, by a mixture model with common components for the classes with the restriction that equal weight is put on all components, $w_{il} = p(g_i|c_l) = K \cdot p(g_i)p(c_l|g_i)$.

Parameter estimation is often proposed to be done by minimizing the quadratic loss function (5), but this is seldom done in practice because this nonlinear optimization is regarded to be to hard and the use of gradient descent method is found to have low efficiency, Xu et al. (1994). Instead the parameter estimation is often done sequentially. The scale matrices are given ad hoc values and estimates of the locations are found using a clustering method, as in Moody and Darken (1989), or selected (randomly or stepwise) from all observation in the training set. The weights are then estimated minimizing the quadratic loss function, that is, by least squares.

The main strength of the MC^2 framework lies in its flexibility in method and model choice with the basis in the class-conditional mixture models and also in its near connections to the methods mentioned above. We will return with further comments in section 5.

3 Model Selection

We will here look at different choices for component distributions and at choosing the component architecture.

3.1 Choice of Component Distributions

One century has passed since Karl Pearson in 1894 published his article on parametric estimation in a mixture of two univariate Gaussian component densities with equal variances using the method of moments. There has since then been a considerable amount of work in the area of mixture models. Especially the past decade has witnessed substantial developments in the application of statistical models based on mixture distributions.

From the previous section we have seen that two slightly different finite mixture models are considered for representing the class-conditional distributions; (1) for common component densities for the classes and (2) for class specific component densities. The primary use for these finite mixture models are as mathematical devices.

The component distributions can be discrete or continuous and there is in general no requirement that the component distributions should belong to the same parametric family. However, we will here restrict us to considering continuous component densities from the same parametric family.

If the probability density function of a random variable $\boldsymbol{X}$ can be written as $f(\boldsymbol{x}) = h(\boldsymbol{x}^T\boldsymbol{x})$ the distribution of $\boldsymbol{X}$ is said to be spherically symmetric, see Mardia et al. (1979). For spherically symmetric densities the equiprobability contours are hyperspheres. Elliptically symmetric densities are densities of the form

$$f(\boldsymbol{x}) = h((\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))$$

where $\boldsymbol{\mu}$ is a location vector and $\boldsymbol{\Sigma}$ is a scale matrix. If $\boldsymbol{X}$ has a elliptically symmetric density the transformed variable $\boldsymbol{Y} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{X}-\boldsymbol{\mu})$ has a spherically symmetric density. Spherically symmetric densities are also called radial symmetric by for example Moody and Darken (1989).

We will here focus on two classes of unimodal elliptically symmetric densities, that is, densities that have a single maximum centred at the location vector. The two classes of densities considered are the Gaussian density and the multivariate Student-t density.

3.1.1 Gaussian Component Densities

The use of Gaussian component densities is a popular choice in mixture modelling,

$$f(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{|\boldsymbol{\Sigma}^{-1}|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}}\exp[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})].$$

The Gaussian distribution is a member of the exponential family. The loglikelihood for a mixture of exponential family densities can be unbounded and may have many local optima, saddle points and singularities, see Redner and Walker (1984) and section 4.3 of Titterington et al. (1985). When estimating the parameters of a mixture we will aim at choosing the parameters corresponding to the largest local maximum of the loglikelihood, and thus avoid the parameter choices for which the loglikelihood is unbounded. We will here define the maximum likelihood estimates as the choice of parameters for which the loglikelihood attains its largest local maximum, as done in Redner and Walker (1984).

We can find the maximum likelihood estimates by solving a set of likelihood equations. The likelihood equations based on the joint loglikelihood (3) for class-conditional Gaussian mixtures are found in section 3.4 of Langaas (1995b). These likelihood equations do not have an explicit solution, but they can be the basis for an alternating routine for estimation, the Expectation Maximization (EM) algorithm of Dempster et al. (1977). The EM algorithm is presented in section 3.5 of Langaas (1995b) for the MC^2 models with common and class specific component distributions. The EM algorithm will under certain regularity conditions converge to a stationary point of the loglikelihood provided that the loglikelihood is bounded, see Wu (1983). Practical experiences from using the EM algorithm for mixture models concludes that in most cases sensible local maxima of the loglikelihood are found, Titterington et al. (1985) section 4.3, provided that the starting points of the algorithm are chosen away from the boundary of the parameter space. See section 3.5 of Langaas (1995b) for more comments on the EM algorithm for Gaussian mixtures.

We will in this report use the EM algorithm for estimation of the parameters and weights, $\boldsymbol{\Theta}$, of the likelihood (3) for class-conditional Gaussian mixtures with common and class specific components. The use of the EM algorithm in the maximization of the conditional likelihood (4) is not found to be as natural and straightforward as for the joint likelihood, and for the conditional likelihood a simplex method is used. In both methods the choice of multiple starting points are important to find a desirable local maximum of the appropriate likelihood.

3.1.2 Multivariate Student-t Component Densities

Modelling a data set by a Gaussian distribution the estimation of the mean is known to be vulnerable to outliers. The multivariate Student-t family of distributions can provide a useful extension of the Gaussian family in this situation, see for example Lange et al. (1989). By replacing the Gaussian distribution with the multivariate Student-t distribution as the choice of component distribution in the class-conditional mixture model, we hope to get a model that is more robust to outliers. The degree of robustness is controlled by the degrees of freedom parameter in the multivariate Student-t distribution. We also hope that the Student-t component density might be suitable for treating small sample situations, and that fewer components are needed than for the Gaussian case.

One Component

The multivariate Student-t density dates back to Cornish (1954) and Dunnet and Sobel (1954) and is presented in Johnsen and Kotz (1972) and in Mardia et al. (1979), page 57,

$$f(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},\nu) = \frac{|\boldsymbol{\Sigma}^{-1}|^{\frac{1}{2}}\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{1}{2})^d\Gamma(\frac{\nu}{2})\nu^{\frac{d}{2}}}[1+\frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})]^{-\frac{\nu+d}{2}}$$

$$E(\boldsymbol{X}) = \boldsymbol{\mu} \;\; (\nu > 1) \quad \textit{and} \quad Cov(\boldsymbol{X}) = E((\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^T) = \frac{\nu}{\nu-2}\boldsymbol{\Sigma} \;\; (\nu > 2).$$

Here d is the dimension of the random variable $\boldsymbol{X}$ and ν is the degrees of freedom. We write $\boldsymbol{X} \sim T_d(\boldsymbol{\mu},\boldsymbol{\Sigma},\nu)$ We will in the following assume that the degrees of freedom is known and that $\nu > 2$.

Estimators for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be found using maximum likelihood estimation. Suppose that we have a training set $\mathcal{T} = (\boldsymbol{x}_j)_{j=1}^n$ of n independent observations from the Student-t distribution. Based on the training set we want to maximize the likelihood of the observations, $\prod_{j=1}^n p(\boldsymbol{x}_j)$. The loglikelihood is given as,

$$l(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu|\{\boldsymbol{x}\}_{j=1}^n) = -\sum_{j=1}^n \frac{\nu+d}{2}[1 + \frac{1}{\nu}(\boldsymbol{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu})] + \frac{n}{2}\log|\boldsymbol{\Sigma}^{-1}| + C(\nu, d) \quad (8)$$

where $C(\nu, d)$ is a function of the degrees of freedom, ν, and the dimension of the sample space, d.

Let D_j denote the squared (scaled) Mahalanobis distance; $D_j = (\boldsymbol{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu})$. Differentiating the Student-t log-density with respect to the location $\boldsymbol{\mu}$ we get,

$$\frac{\partial \log f(\boldsymbol{x}_j|\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}} = \frac{\nu+d}{\nu+D_j}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}) \ \ (j = 1, ..., n).$$

Let $\boldsymbol{\mu}(p)$ denote the pth element of the vector $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}(p,q)$ denote element (p,q) of the covariance matrix $\boldsymbol{\Sigma}$ and $I\boldsymbol{\Sigma}(p,q)$ denote element (p,q) of the the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$. Then we have,

$$\frac{\partial \log f(\boldsymbol{x}_j|\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial I\boldsymbol{\Sigma}(p,q)} = (1 - \frac{\delta_{pq}}{2})[\boldsymbol{\Sigma}(p,q) - \frac{\nu+d}{\nu+D_j}(\boldsymbol{x}_j(p) - \boldsymbol{\mu}(p))(\boldsymbol{x}_j(q) - \boldsymbol{\mu}(q))] \ \ (j = 1, ..., n)$$

where δ_{pq} is the Kronecker delta function.

Based on differentiating the loglikelihood (8) with respect to the location and scale parameters and setting equal to zero we find following likelihood equations,

$$\boldsymbol{\mu} = \frac{\sum_{j=1}^n \frac{\nu+d}{\nu+D_j}\boldsymbol{x}_j}{\sum_{j=1}^n \frac{\nu+d}{\nu+D_j}}$$

and

$$\boldsymbol{\Sigma} = \frac{1}{n}\sum_{j=1}^n \frac{\nu+d}{\nu+D_j}(\boldsymbol{x}_j - \boldsymbol{\mu})(\boldsymbol{x}_j - \boldsymbol{\mu})^T.$$

These likelihood equations are given in Maronna (1976) and Lange et al. (1989). Comparing the above likelihood equations with the likelihood equations in the Gaussian case; $\boldsymbol{\mu} = \frac{1}{n}\sum_{j=1}^n \boldsymbol{x}_j$ and $\boldsymbol{\Sigma} = \frac{1}{n}\sum_{j=1}^n(\boldsymbol{x}_j - \boldsymbol{\mu})(\boldsymbol{x}_j - \boldsymbol{\mu})^T$, the Student-t likelihood equations include a weighting term $w_j = \frac{\nu+d}{\nu+D_j}$ dependent on the squared Mahalanobis distance and therefore also on the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. In the limit as the degrees of freedom goes to infinity, $\nu \to \infty$, the weights approach unity, $w_j \to 1$, and we have the Gaussian solution. The weighting term down-weights observations far away from the location $\boldsymbol{\mu}$.

Kent et al. (1994) give conditions for when there exists a unique solution to the likelihood equations. These conditions are as follows. "Suppose $\{\boldsymbol{x}_j\}_{j=1}^n$ is a data set in $\mathbb{R}^d$, the degrees of freedom is $\nu > 1$, the number of observations are $n \geq d+1$ and all subsets of $d+1$ points span a hyperplane of dimension d. Then there exists a unique solution $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ to the above likelihood equations which maximizes the loglikelihood (8)." These results are based on identifying the maximum likelihood estimates of $T_d(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$, with ν known, with the maximum likelihood estimates of $T_{d+1}(\mathbf{0}, \mathbf{\Sigma}, \nu - 1)$ for $\nu > 1$, see Kent et al. (1994) for further details.

Estimation of the parameters of the Student-t distribution can be carried out iteratively using the EM algorithm of Dempster et al. (1977). We will here only consider the case when the degrees of freedom is known. The EM algorithm is outlined in appendix B (for the case when the degrees of freedom, ν, is unknown).

The EM algorithm proceeds as follows. Given estimates $\boldsymbol{\mu}^{(s)}$ and $\mathbf{\Sigma}^{(s)}$ at the sth iteration step, the weights $w_j^{(s)}$ are calculated using the current estimates of the parameters.

$$\begin{aligned} D_j^{(s)} &= (\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)})^T (\mathbf{\Sigma}^{(s)})^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)}) \;\; (j = 1, ..., n) \\ w_j^{(s)} &= \frac{\nu + d}{\nu + D_j^{(s)}} \;\; (j = 1, ..., n) \end{aligned}$$

Then calculate new estimates $\boldsymbol{\mu}^{(s+1)}$ and $\mathbf{\Sigma}^{(s+1)}$.

$$\begin{aligned} \boldsymbol{\mu}^{(s+1)} &= \frac{\sum_{j=1}^n w_j^{(s)} \boldsymbol{x}_j}{\sum_{j=1}^n w_j^{(s)}} \\ \mathbf{\Sigma}^{(s+1)} &= \frac{1}{n} \sum_{j=1}^n w_j^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}^{(s+1)})(\boldsymbol{x}_j - \boldsymbol{\mu}^{(s+1)})^T \end{aligned}$$

Here the EM algorithm takes the form of an iterative reweighted least squares algorithm (IRLS), as pointed out in section 4.6 of Dempster et al. (1977) and in Rubin (1983). The IRLS is often associated with robust estimation and M-estimation, as in Maronna (1976), section 8.6 of Huber (1981) and chapter 2 of Hampel et al. (1986). When M-estimates are maximum likelihood estimates, the associated IRLS algorithm is an EM algorithm, Rubin (1983). Based on the relationship between the maximum likelihood estimates of $T_d(\boldsymbol{\mu}, \mathbf{\Sigma}, \nu)$ and $T_{d+1}(\mathbf{0}, \mathbf{\Sigma}, \nu - 1)$ a slightly different EM algorithm, to the one presented here, is supplied in Kent et al. (1994).

Arslan et al. (1993) have studied the convergence of the EM algorithm for univariate Student-t distributions with known scale ($\sigma^2 = 1$) when the degrees of freedom is small ($\nu = 0.1$ and $\nu = 0.05$ in their simulation studies). They found that when the degrees of freedom is *small enough* the loglikelihood can have n local maxima, one close to each sample point. The domain of convergence of a local maximum is a set of starting point from which the EM algorithm converges to that local maximum. Arslan et al. (1993) found examples when a convergence domain consists of a number of disjoint open subsets of the parameter space. Despite the artificial nature of their examples ($\nu = 0.1$, $\nu = 0.05$ and $n = 4$), Arslan et al. (1993) conclude that emphasis must be put on choosing different staring points for the Student-t EM algorithm in order to avoid possible local maxima. (Kent et al. (1994), however, state that when the maximum likelihood estimates exist and are the unique solution to the likelihood equations, then the EM algorithm is guaranteed to converge to the maximum likelihood estimates from an arbitrary starting point.)

The treatment in Arslan et al. (1993) have shown that the univariate Student-t loglikelihood can have multiple local maxima for small values of the degrees of freedom and for small data sets. In the present study we choose the degrees of freedom to be $\nu = 4$, and have used the sample mean as the initial estimate of the location $\boldsymbol{\mu}$ in the EM algorithm. We have not routinely chosen multiple starting point for the EM algorithm for the one component Student-t distribution.

Class-Conditional Mixture with Student-t Component Distributions

We now consider the case when the component distributions of class-conditional mixtures are multivariate Student-t distributions. As for the Gaussian mixture the loglikelihood for a mixture of Student-t component distributions might experience multiple local maxima and the loglikelihood can be unbounded when the component distributions are allowed to be degenerated around observed points. When estimating the parameters of a Student-t mixture we will aim at choosing the parameters corresponding to the largest local maximum of the loglikelihood and to avoid parameter values on the boundary of the parameter space.

The likelihood equations based on the joint likelihood (3) for a class-conditional mixture are found in section 3.3 of Langaas (1995b). For common component distributions for the classes the likelihood equations are found in equations (11)-(15) and for class specific component distributions in equations (16)-(21) of Langaas (1995b).

We will consider parameter estimation in class-conditional Student-t mixtures using the EM algorithm. Applying the EM algorithm results in an iterative scheme similar to that for the one component multivariate Student-t distribution and the Gaussian mixtures. Iterative schemes are presented for common and class specific components below. We assume that the degrees of freedom, ν, is known. The EM algorithm for class-conditional mixtures of Student-t distributions is outlined in appendix B, where the case when ν is unknown also is treated. The multivariate Student-t distribution is not a member of the exponential family and the convergence results for the EM algorithm for mixtures of exponential family distributions found in Redner and Walker (1984) do not apply. Further research is needed to theoretically establish the local convergence of a mixture of multivariate Student-t distributions.

As shown below the EM algorithm for the class-conditional multivariate Student-t mixture bears great resemblance to the EM algorithm for the class-conditional Gaussian mixture. The only difference is an additional weight put on the observations, equivalent to the weight used for the one component multivariate Student-t distribution. We will in this report use the EM algorithm for estimation of the parameters and weights, $\boldsymbol{\Theta}$, of the likelihood (3) for class-conditional multivariate Student-t mixtures with common and class specific components. For the conditional likelihood (4) a simplex method is used. As for the Gaussian mixture the choice of multiple starting points are important to find a desirable local maximum of the appropriate likelihood. Empirical results for maximizing the joint likelihood (3) using the class-conditional Student-t mixture have shown that a larger number of EM steps is needed to converge than for the Gaussian mixture (about twice as many using the same convergence criterion). No difference in the convergence speed is found for the simplex method for Gaussian and Student-t mixtures.

Common Components The prior probabilities, $p(g_i)$, for the classes are estimated using the proportions of each class in the training set, which is the maximum likelihood estimator.

1. $T_d(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l, \nu)$; we start with the case of separate covariance matrices for each component density. At the sth step of the algorithm we have estimates $\boldsymbol{\mu}_l^{(s)}$, $\boldsymbol{\Sigma}_l^{(s)}$ and $p(c_l|g_i)^{(s)}$ for $i = 1, ..., M$ and $l = 1, ..., K$. From these we calculate

$$\begin{aligned}
D_{jl}^{(s)} &= (\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})^T (\boldsymbol{\Sigma}_l^{(s)})^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)}) \;\; (j = 1, ..., n; l = 1, ..., K) \\
p(c_l|g_i, \boldsymbol{x}_j)^{(s)} &= \frac{p(c_l|g_i)^{(s)} f_l(\boldsymbol{x}_j|\boldsymbol{\mu}_l^{(s)}, \boldsymbol{\Sigma}^{(s)}, \nu)}{\sum_{m=1}^{K} p(c_m|g_i)^{(s)} f_m(\boldsymbol{x}_j|\boldsymbol{\mu}_m^{(s)}, \boldsymbol{\Sigma}^{(s)}, \nu)} \\
&\qquad (j = 1, ..., n; i = 1, ..., M; l = 1, ..., K) \\
w_{jil}^{(s)} &= p(c_l|g_i, \boldsymbol{x}_j)^{(s)} \cdot \frac{\nu + d}{\nu + D_{jl}^{(s)}} \;\; (j = 1, ..., n; i = 1, ..., M; l = 1, ..., K)
\end{aligned}$$

We then find new estimates at the $(s + 1)$th step. Here n_i denotes the number of observations from class g_i and $z_{ji} = 1$ if $\boldsymbol{x}_j \in g_i$ and zero otherwise.

$$\begin{aligned}
\boldsymbol{\mu}_l^{(s+1)} &= \frac{\sum_{j=1}^{n} \sum_{i=1}^{M} z_{ji} w_{jil}^{(s)} \boldsymbol{x}_j}{\sum_{j=1}^{n} \sum_{i=1}^{M} z_{ji} w_{jil}^{(s)}} \;\; (l = 1, ..., K) \\
\boldsymbol{\Sigma}_l^{(s+1)} &= \frac{\sum_{j=1}^{n} \sum_{i=1}^{M} z_{ji} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})^T}{\sum_{j=1}^{n} \sum_{i=1}^{M} z_{ji} p(c_l|g_i, \boldsymbol{x}_j)^{(s)}} \;\; (l = 1, ..., K) \\
p(c_l|g_i)^{(s+1)} &= \frac{1}{n_i} \sum_{j=1}^{n} z_{ji} p(c_l|g_i, \boldsymbol{x}_j)^{(s)} \;\; (i = 1, ..., M; l = 1, ..., K)
\end{aligned}$$

2. $T_d(\boldsymbol{\mu}_{il}, \boldsymbol{\Sigma}, \nu)$; this is the case with common covariance matrices for all component densities. The sth step values for $D_{jl}^{(s)}$, $p(c_l|g_i, \boldsymbol{x}_j)^{(s)}$ and $w_{jil}^{(s)}$ include $\boldsymbol{\Sigma}^{(s)}$ instead of $\boldsymbol{\Sigma}_l^{(s)}$ and the $(s + 1)$th step estimate for $\boldsymbol{\Sigma}$ is given below.

$$\boldsymbol{\Sigma}^{(s+1)} = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{M} \sum_{l=1}^{K} z_{ji} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})^T$$

Class Specific Components Also here the prior probabilities, $p(g_i)$, for the classes are estimated using the proportions of each class in the training set.

1. $T_d(\boldsymbol{\mu}_{il}, \boldsymbol{\Sigma}_{il}, \nu)$; we start with the case of separate covariance matrices for each component density. At the sth step of the algorithm we have estimates $\boldsymbol{\mu}_{il}^{(s)}$, $\boldsymbol{\Sigma}_{il}^{(s)}$ and $p(c_{il}|g_i)^{(s)}$ from which we calculate

$$\begin{aligned}
D_{jil}^{(s)} &= (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T (\boldsymbol{\Sigma}_{il}^{(s)})^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)}) \;\; (j = 1, ..., n; i = 1, ..., M; l = 1, ..., k_i) \\
p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)} &= \frac{p(c_{il}|g_i)^{(s)} f_{il}(\boldsymbol{x}_j|\boldsymbol{\mu}_{il}^{(s)}, \boldsymbol{\Sigma}_{il}^{(s)}, \nu)}{\sum_{m=1}^{K} p(c_{im}|g_i)^{(s)} f_{im}(\boldsymbol{x}_j|\boldsymbol{\mu}_{im}^{(s)}, \boldsymbol{\Sigma}_{im}^{(s)}, \nu)} \\
&\qquad (j = 1, ..., n; i = 1, ..., M; l = 1, ..., k_i) \\
w_{jil}^{(s)} &= p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)} \cdot \frac{\nu + d}{\nu + D_{jil}^{(s)}} \;\; (j = 1, ..., n; i = 1, ..., M; l = 1, ..., k_i)
\end{aligned}$$

Then new estimates are found at the $(s+1)$th step.

$$\begin{aligned}
\boldsymbol{\mu}_{il}^{(s+1)} &= \frac{\sum_{j=1}^{n} z_{ji} w_{jil}^{(s)} \boldsymbol{x}_j}{\sum_{j=1}^{n} z_{ji} w_{jil}^{(s)}} \quad (i = 1, ..., M; l = 1, ..., k_i) \\
\boldsymbol{\Sigma}_{il}^{(s+1)} &= \frac{\sum_{j=1}^{n} z_{ji} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T}{\sum_{j=1}^{n} z_{ji} p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)}} \quad (i = 1, ..., M; l = 1, ..., k_i) \\
p(c_{il}|g_i)^{(s+1)} &= \frac{1}{n_i} \sum_{j=1}^{n} z_{ji} p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)} \quad (i = 1, ..., M; l = 1, ..., k_i)
\end{aligned}$$

2. $T_d(\boldsymbol{\mu}_{il}, \boldsymbol{\Sigma}_i, \nu)$; we continue with the case of class specific covariance matrices. The sth step values for $D_{jil}^{(s)}$, $p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)}$ and $w_{jil}^{(s)}$ include $\boldsymbol{\Sigma}_i^{(s)}$ instead of $\boldsymbol{\Sigma}_{il}^{(s)}$ and the $(s+1)$th step estimate, $\boldsymbol{\Sigma}_i^{(s+1)}$, is given below.

$$\boldsymbol{\Sigma}_i^{(s+1)} = \frac{1}{n_i} \sum_{j=1}^{n} z_{ji} \sum_{l=1}^{k_i} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T \quad (i = 1, ..., M)$$

3. $T_d(\boldsymbol{\mu}_{il}, \boldsymbol{\Sigma}, \nu)$; finally we turn to the case of common covariance matrices. As before, the sth step values for $D_{jil}^{(s)}$, $p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)}$ and $w_{jil}^{(s)}$ include $\boldsymbol{\Sigma}^{(s)}$ instead of $\boldsymbol{\Sigma}_{il}^{(s)}$.

$$\boldsymbol{\Sigma}^{(s+1)} = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{M} z_{ji} \sum_{l=1}^{k_i} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T$$

3.1.3 Examples

A small study has been conducted to compare the performance of Gaussian and Student-t component densities with four degrees of freedom, $\nu = 4$. The study is based on one synthetic data set of Gaussian mixtures (SGM) and one real data set of forensic glass. The SGM data set consists of three classes, each class is an equal mixture of two Gaussian component densities with class specific diagonal matrices of covariance, see Langaas (1995a). The forensic glass data set is collected by B. German and presented in Ripley (1994). Both data sets are analysed and discussed in Langaas (1995b).

The estimation and classification results have been divided into four categories according to method (joint or conditional likelihood) and model (class specific or common components). The results using Gaussian and Student-t component densities are compared within each category. Within each category a selection of component architectures and all choices of covariance matrices (common or specific to the classes or components, and full, diagonal or diagonal with constant variances) are fitted. For convenience only the error rate for the model giving the lowest error rate on the test set is quoted within each category. The results of the fitted MC^2 models are found in table 1.

The synthetic Gaussian mixtures data set is constructed by drawing from a mixture of class specific Gaussian component densities with component architecture (2,2,2) and class specific diagonal covariance matrices. Even so, the use of Student-t component densities, with four

Comparative Study of Gaussian and Student-t Component Distributions Error Rates for Test Sets							
MC^2	component	Class specific components			Common components		
	density	error	K	model	error	K	model
Synthetic Gaussian Mixtures (test set of size 600)							
JOINT	Gaussian	12.0 (1.3)	(2,2,2)	$N(\boldsymbol{\mu}_{il}, \boldsymbol{\Sigma}_i)$	11.8 (1.3)	12	$N(\boldsymbol{\mu}_l, \sigma_l^2 \boldsymbol{I})$
JOINT	Student-t	11.8 (1.3)	(2,2,2)	$T(\boldsymbol{\mu}_{il}, \boldsymbol{\Sigma}_i)$	11.3 (1.3)	6	$T(\boldsymbol{\mu}_l, \sigma_l^2 \boldsymbol{I})$
COND	Gaussian	11.2 (1.3)	(4,4,4)	$N(\boldsymbol{\mu}_{il}, \sigma^2 \boldsymbol{I})$	11.5 (1.3)	6	$N(\boldsymbol{\mu}_l, \sigma_l^2 \boldsymbol{I})$
COND	Student-t	12.2 (1.3)	(1,1,1)	$T(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}_i))$	11.7 (1.3)	6	$T(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}))$
Forensic Glass - Full Classification (test set of size 96)							
JOINT	Gaussian	39.6 (5.0)	(2,2,2,1)	$N(\boldsymbol{\mu}_{il}, \boldsymbol{\Sigma})$	34.4 (4.8)	8	$N(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}))$
JOINT	Student-t	31.3 (4.7)	(2,2,2,1)	$T(\boldsymbol{\mu}_{il}, \sigma_i^2 \boldsymbol{I})$	37.5 (4.9)	6	$T(\boldsymbol{\mu}_l, \sigma_l^2 \boldsymbol{I})$
COND	Gaussian	32.3 (4.8)	(2,2,2,1)	$N(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$	29.2 (4.6)	4	$N(\boldsymbol{\mu}_l, \sigma_l^2 \boldsymbol{I})$
COND	Student-t	31.3 (4.7)	(1,1,1,2)	$T(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}_i))$	27.0 (4.5)	7	$T(\boldsymbol{\mu}_l, \sigma_l^2 \boldsymbol{I})$
Forensic Glass - Confusion between WNF and WF allowed (test set of size 96)							
JOINT	Gaussian	14.4 (3.6)	(2,2,2,1)	$N(\boldsymbol{\mu}_{il}, \sigma^2 \boldsymbol{I})$	13.5 (3.4)	6	$N(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}))$
JOINT	Student-t	15.6 (3.7)	(2,2,2,2)	$T(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$	14.6 (3.6)	6	$T(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}))$
COND	Gaussian	11.5 (3.3)	(2,2,2,1)	$N(\boldsymbol{\mu}_{il}, \sigma^2 \boldsymbol{I})$	11.5 (3.3)	4	$N(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}))$
COND	Student-t	16.7 (3.8)	(1,1,1,1)	$T(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$	11.5 (3.3)	7	$T(\boldsymbol{\mu}_l, \sigma_l^2 \boldsymbol{I})$

Table 1: Error rates (%) for MC^2 models with Gaussian and Student-t component densities (with four degrees of freedom, $\nu = 4$). The columns denoted "error" give the error rate on the test set together with an approximate standard deviation for the error rate in parenthesis. The standard deviation is based on assuming that the number of misclassified observations is binomially distributed. The columns denoted "K" give the component architecture and the columns denoted "model" give the covariance structure used. The covariance matrices can be full, diagonal or diagonal with constant variances and can be specific to each component or class or common to all classes or components. JOINT denotes that the joint likelihood (3) and COND that the conditional likelihood (4) is used. The synthetic Gaussian mixtures data set has two explanatory variables ($d = 2$) and three classes ($M = 3$). Both the training set and the test set is of size 600. For the synthetic Gaussian mixtures a grid search is performed for the architectures K={(1,1,1),(2,2,2),(3,3,3),(4,4,4)} for class-specific components and K={3,6,9,12} for common components for the classes. Pairwise comparisons for the synthetic Gaussian mixtures data set have a 95 % confidence interval of half-length about 0.8 %. The forensic glass data set has nine explanatory variables ($d = 9$) and has a training set of size 89 and a test set of size 96. For full classification there are four classes ($M = 4$). When confusion between two types of glass is allowed, window nonfloat (WNF) and window float (WF), these errors are assumed to have zero loss and are not counted. For the forensic glass data set the architectures investigated are K={(1,1,1,1),(1,1,1,2),(1,1,2,2,),(2,2,2,1),(2,2,2,2)} for class-specific components and K={4,5,6,7,8} for common components for the classes. Pairwise comparisons for the forensic data set have a 95 % confidence interval of half-length about 5%.

degrees of freedom, produces very good results. The use of Student-t component densities instead of Gaussian component densities seems to have stronger impact for the smaller forensic glass data set (89 observations in the training set and 96 observations in the test set, 9 variables) than for the large synthetic Gaussian mixtures data set (600 observation for each of the training and the test set, 2 variables). For the forensic glass data set the use of Student-t component densities is clearly more suitable than the Gaussian component densities for the situation with common components. The difference in performance is, however, in most cases not significant (on a 5% significance level) using pairwise comparisons. Using unpaired comparisons the difference is not significant.

For the forensic glass data set the results from maximizing the conditional likelihood are better than the results from maximizing the joint likelihood. For the synthetic Gaussian mixtures data set the two optimization criteria give similar results.

3.2 Choice of Component Architecture

The MC^2 framework has been presented with the aim of constructing classifiers. The performance of a classifier can be measured by its error rate. From another point of view the MC^2 framework is based on representing the class-conditional densities by finite mixture models. Both these two perspectives can be taken when trying to find a good component architecture.

At one extreme each class-conditional distribution can be represented by one mixture component, at the other extreme each observation can be chosen as the location of a mixture component. However, we seek the *smallest* number of components that provides a satisfactory model-fit to a given data set. For a class-conditional mixture with class specific components an important question is whether the number of mixture components within each class should be decided separately for each class or simultaneously for all classes. For a class conditional mixture with common components the number of components should at least be as large as the number of classes, $K \geq M$.

3.2.1 Determining the Number of Components of a Finite Mixture

The problem of determining the number of components of a finite mixture is an important and difficult task with a long history. A review of different approaches to solving the problem is found in chapter 1.10 of McLachlan and Basford (1988), but the problem is still not completely resolved. Two branches of the research are concentrated with the assessment of modality and with likelihood ratio tests.

Let us in this section consider choosing the number, K, of unimodal component densities of a finite mixture. If the components of a mixture of unimodal densities are sufficiently well separated there will be one mode for each component. That is, the number of component densities equals the number of modes. For unimodal component densities the number of components should always be larger than or equal to the number of modes and the number of modes can therefore give a lower limit on the number of components. However, this may of cause not be the case when regarding a sample from a distribution, for example, bimodality can occur in a sample from a single Gaussian distribution. In the univariate case the dip test can be used to test a unimodal versus a multimodal distribution. The dip test measures the maximum difference between the empirical distribution function and the unimodal distribution function chosen to minimize that maximum difference, see Panel on Discriminant Analysis, Classification and Clustering (1989), but has no simple generalization to more than one dimension.

Gath and Geva (1989) use the fuzzy K-means and fuzzy maximum likelihood algorithm in two sequential steps to perform clustering. They iterate the scheme for increasing number of clusters and for each run compute three performance measures. One of these performance measures is the fuzzy hypervolume, which is the sum of the square root of the determinant of the covariance matrices. Gath and Geva (1989) found that the fuzzy hypervolume gave good results on data from a mixture of Gaussian distributions. Wong (1985) consider bootstrap test procedures for investigating the modality of general univariate distribution based on a clustering method (called kth nearest neighbour clustering).

Clustering is in general performed by optimizing a criterion function. To find a good number of clusters the clustering procedure can be repeated for different number of clusters to see how the criterion function changes, see Duda and Hart (1973). Some criterion functions will decrease/increase monotonically with the number of clusters. This is also the case for most the MC^2 optimization criteria of section 2.1 as functions of the number of component densities. To avoid overfitting, these criteria functions must be modified if they are used to find the optimal number of component densities. One solution is to include a penalty on the number of components. This introduces the problem of finding a useful form for the penalty term. For cluster analysis Peck et al. (1989) use a linear term in the number of components, K, as the penalty, λK (which leads to the problem of choosing λ). Let $C(\mathbf{\Theta}|\mathcal{T})$ denote the penalized criterion function,

$$C(\mathbf{\Theta}|\mathcal{T}) = \mathcal{J}(\mathbf{\Theta}|\mathcal{T}) + P(K) = \mathcal{J}(\mathbf{\Theta}|\mathcal{T}) + \lambda K$$

Here $\mathcal{J}(\mathbf{\Theta}|\mathcal{T})$ is the original criterion and the term $P(K)$ is a penalty term for the cost of having K components. Peck et al. (1989) find bootstrap confidence intervals for the number of components using the above linear penalty on the number of components when the data is from a mixture of Gaussian densities.

An obvious way to approach the problem of determining the number of mixture components is to use the likelihood ratio test for the null hypothesis $H_0 : K = k_0$ versus the alternative hypothesis $H_1 : K = k_1$ components. However, since the loglikelihood of the mixture can be unbounded the regularity conditions required for the loglikelihood ratio $-2\log\gamma$ to have a χ^2-distribution is not fulfilled, see chapter of 1.10 McLachlan and Basford (1988). In the simplest case when $k_0 = 1$ and $k_1 = 2$ this can be motivated by considering the equivalent test of whether the probability for component one is unity, $H_0 : p(c_1) = 1$. Since this value of the component probability is on the boundary of the parameter space the standard regularity conditions are not fulfilled. McLachlan (1987) consider bootstrapping the loglikelihood test statistic for Gaussian mixtures. Under H_0 parametric bootstrap samples are generated from the mixture density with k_0 component densities inserted the maximum likelihood estimate of the parameters of the mixture model under the null hypothesis from the original sample. Mixture models with k_0 and k_1 components are fitted, by maximum likelihood estimation, to each sample in turn to construct the loglikelihood test statistic. From these, say B, bootstrap samples the rejection or retension of the null hypothesis at the approximate significance level α can be done using a Monte Carlo test, see for example section 7 of Ripley (1987), rejecting the null hypothesis if $-2\log\gamma$ for the original data is greater than the bth smallest of the B bootstrap test statistics where b is chosen so that $\alpha = \frac{B-b+1}{B+1}$. Furman and Lindsay

(1994) have developed two procedures for testing the number of components of an univariate mixture of Gaussian components with equal variances based on moment estimators. One of the procedures is to replace the maximum likelihood estimators in McLachlan (1987) by moment estimators. Roeder (1994) proposes a graphical technique and a test for determining the number of components in a univariate mixture of Gaussian components.

3.2.2 Stepwise Component Architecture Selection

We will in this section focus on the error rate in order to select the "best" component architecture.

The classification is assumed to be based on the posterior probabilities for the classes with the potential of using different costs for each misclassification. We define the *misclassification badness* to be the vector of weighted error rates for the classes and the *total misclassification badness* to be the sum of the misclassification badnesses. The weights are chosen (by the data analyst) to reflect the cost of misclassification for each class.

To construct a component architecture selection routine it is necessary to give some guidelines about what is meant by the "best" component architecture. The "best" component architecture will be a small architecture that gives a low total misclassification badness on both the training set and on a separate validation set. Further, the total misclassification badness will be of as close as possible magnitude for the training set and the validation set. Apart from these guidelines, subjective evaluation is needed in concrete cases.

For the class specific component situation we present a stepwise algorithm that is based on both the total misclassification badness and the misclassification badnesses for each class. For the situation with common components, we suggest to perform a grid search, that is, fit a set of component architectures and then choose the "best" component architecture.

Stepwise Component Architecture Selection for Class Specific Components The stepwise routine starts by defining a lower and a upper component architecture limit. The lower limit vector, K_L, is usually set at one component for each class and the upper limit vector, K_U, at the maximum number of components that the data analyst is willing to choose. The routine needs both a training set and a validation set. A component architecture is said to be *valid* if it is larger than or equal to the lower limit, K_L, and smaller than or equal to the upper limit, K_U. The algorithm selects a set of *transit* architectures from which the best architecture is chosen.

The selection procedure proceeds as follows.

Initial State The selection procedure starts at the smallest possible component architecture defined or at $K_L = (1, 1, ..., 1)$. The chosen MC^2 model is fitted and the misclassification badnesses for the training and the validation set are calculated.

Forward Step a) Rank the classes according to their misclassification badnesses on the training set. Increase the number of components in the class, g_i, with the largest misclassification badness in the training set by one, $k_i = k_i + 1$, and proceed to b). If this class lost one component in the previous step the architecture is not accepted. Then goto c).

b) If the chosen component architecture is valid, fit the new component architecture and calculate the misclassification badnesses for the training and the validation set. If the total misclassification badness both on the training and the validation set has not deteriorated, accept this as the new transit architecture. If this is not the case accept with probability equal to the total error rate on the training set. (This is done to avoid getting stuck in local optima.) If the new architecture is accepted continue to the backward step. If the new architecture is not accepted, go to c).

c) Increase the number of components in the class next in the misclassification badness ranking, then goto b). If there are no more classes to be tried continue to the backward step.

Backward Step a) Rank the classes according to their misclassification badnesses on the training set. Decrease the number of components in the class with the lowest misclassification badness on the training set by one and proceed to b). If this class has just received one new component this architecture is not accepted, goto c).

b) If the chosen component architecture is valid, fit the new model and calculate the misclassification badnesses for the training and the validation set. If the total misclassification badness both on the training and the validation set has not deteriorated, accept this as the new transit architecture. If this is not the case accept with probability equal to the total misclassification rate on the training set. If the new architecture is accepted continue to the closing step. If not, go to c).

c) Decrease the number of components in the class next in the misclassification badness ranking and goto b). If there are no more classes to be tried go on to the closing step.

Closing Step If no forward selection or backward elimination has been accepted the search is ended, else go back to the forward step. If the search is ended the optimal architecture is defined to be the "best" of the transit architectures.

The stepwise routine has been applied to different MC^2 models for the synthetic Gaussian mixtures data set. Classification is done to the class with the highest posterior probability and equal weight is put on the error rate for each class. In Langaas (1995b) the error rates on a test set is quoted for component architectures $K = \{(1,1,1),(2,2,2),(3,3,3),(4,4,4)\}$ for the joint (3) and conditional (4) likelihood criteria. For the case of Gaussian component densities with class specific diagonal covariance matrices a grid search is performed over all component architectures with up to four components in each class using the joint likelihood optimization criterion (3). This resulted in $4^3 = 64$ different component architectures. Component architectures $K = (2,4,1)$ and $K = (3,4,2)$ both gave the lowest error rates on the test set of 11.5%. The architecture $K = (3,4,2)$ had a error rate on the training set of 9.33% and $K = (2,4,1)$ had 9.83%. Other architectures had lower error rate on the training set, but a higher error rate on the test set. The stepwise routine was tried on this MC^2 situation using the test set to perform the duties of a validation set. The stepwise routine tried 8 component architectures (accepting four as transit architectures) and found $K = (2,4,1)$ to be the optimal component architecture. This stepwise component architecture selection routine will be used further in the subsequent sections.

Other solutions to the problem of specifying the component architecture is to use subject matter knowledge or heuristic arguments. In the implementation of the Learning Vector Quantizer (LVQ) in Kohonen et al. (1992) the same number of component densities is chosen initially within each class. The number of component densities are found as $\min(round(0.4 * M * (M - 1 + d/2), 0), n)$ where M is the number of classes, n the number of observation and d the dimension of the input space. The proportion of codewords in each class is altered so that the median of the nearest neighbour distances between codebooks within a class is roughly similar across the classes.

4 Parameter Estimation and Method Evaluation

In section 4.1 we compare the MC^2 optimization criteria based on simulation studies. In section 4.2 we look at parametric penalties for the MC^2 estimation and in section 4.3 a medical data set is analysed.

4.1 Comparing Optimization Criteria

The emphasis will be on comparing the joint and the conditional likelihood criteria.

If the underlying class-conditional distributions are known, then maximizing the joint likelihood (3) will result in maximum likelihood estimators which asymptotically will be more efficient than the estimators obtained from maximizing the conditional likelihood (4). In most cases, however, the underlying class-conditional distributions are unknown.

4.1.1 Joint versus Conditional Likelihood, Theory

In order to gain some insight and for comparison we will in the following outline the MAP class boundary and the joint and conditional likelihood for univariate Gaussian and Student-t class-conditional distributions for the situation with two classes.

Class-conditional Gaussian Distributions Consider the following class-conditional distributions,

$$X|g_1 \sim N(\mu_1, \sigma^2) \;\; and \;\; X|g_2 \sim N(\mu_2, \sigma^2)$$

with prior probabilities $p(g_1)$ and $p(g_2)$, respectively. The MAP rule for classification defines the class boundary between class g_1 and g_1 at X_B satisfying $p(g_1|X_B) = p(g_2|X_B)$. This leads to the following class boundary,

$$X_B = \frac{\mu_1 + \mu_2}{2} - \frac{\sigma^2}{\mu_1 - \mu_2} \log \frac{p(g_2)}{p(g_1)}$$

With equal class priors, $p(g_1) = p(g_2)$, the class boundary is the average of the means, $X_B = \frac{\mu_1 + \mu_2}{2}$.

The parameters of this model are $\mathbf{\Theta} = \{\mu_1, \mu_1, \sigma^2, p(g_1)\}$. Based on a data set of n_1 independent observations from class g_1 and n_2 independent observations from class g_2 the joint loglikelihood in this case can be written as

$$l_{joint} = \sum_{j=1}^{n}\sum_{i=1}^{2} z_{ji}[-\frac{1}{2\sigma^2}(x_j-\mu_i)^2] + n_1 \log p(g_1) + n_2 \log(1-p(g_2)) - (n_1+n_2)\log\sigma + c \quad (9)$$

where $z_{ji}=1$ if $x_j \in g_i$ and zero otherwise, $n = n_1 + n_2$ and c is a constant. Maximizing this loglikelihood results in the well known maximum likelihood estimators;

$$\hat{\mu}_i = \frac{1}{n_i}\sum_{j=1}^{n} z_{ji}x_j, \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{2} z_{ji}(x_j-\hat{\mu}_i)^2, \qquad \hat{p}(g_1) = \frac{n_1}{n}$$

The conditional loglikelihood (4) for the class-conditional Gaussians can be written as follows,

$$l_{cond} = \sum_{j=1}^{n}\delta_j(d+bx_j) - \sum_{j=1}^{n}\log(1+\exp(d+bx_j)) \quad (10)$$

where

$$\delta_j = \begin{cases} 0 & \text{if } \boldsymbol{x}_j \in g_1 \\ 1 & \text{if } \boldsymbol{x}_j \in g_2 \end{cases}, \qquad b = \frac{\mu_2-\mu_1}{\sigma^2}, \qquad d = \log\frac{1-p(g_1)}{p(g_1)} - \frac{\mu_2-\mu_1}{\sigma^2}\cdot\frac{\mu_1+\mu_2}{2}$$

This reparameterisation clearly shows that it is not possible to estimate all the parameters $\boldsymbol{\Theta} = \{\mu_1, \mu_1, \sigma^2, p(g_1)\}$. The estimates of b and d are found maximizing (10). According to section 8.5 of McLachlan (1992), the bias of the maximum conditional likelihood estimates of b and d can be of some concern in practice in small samples and he gives expressions for correcting the bias. Using the parameters b and d the class boundary is

$$X_B = -\frac{d}{b}$$

and is estimable. This reparametrisation also shows that the conditional likelihood (10) is more general than the joint likelihood, that is, it is valid for other distributions than the Gaussian, see section 6.4.1 of Seber (1984). The latter does not include the Student-t distribution.

Class-conditional Student-t Distributions If Student-t class-conditional distributions are assumed,

$$X|g_1 \sim T(\mu_1, \sigma^2, \nu) \ \ and \ \ X|g_2 \sim T(\mu_2, \sigma^2, \nu)$$

with probabilities $p(g_1)$ and $p(g_2)$, respectively, the MAP class boundary, X_B, is found as the roots of the second order equation

$$(\alpha-1)\frac{1}{\nu\sigma}X_B^2 - \frac{2}{\nu\sigma^2}(\alpha\mu_1-\mu_2)X_B + (\alpha-1) + \frac{1}{\nu\sigma^2}(\alpha\mu_1^2-\mu_2^2) = 0$$

where $\alpha = \exp(\frac{2}{\nu+1}\log\frac{1-p(g_1)}{p(g_1)})$. With equal class prior, $p(g_1) = p(g_1)$, the class boundary is the average of the means, $X_B = \frac{\mu_1+\mu_2}{2}$, as in the Gaussian case.

The joint loglikelihood based on n_1 independent observations from class g_1 and n_2 independent observations from class g_2 in this case is as follows,

$$l_{joint} = \sum_{j=1}^{n}\sum_{i=1}^{2} z_{ji}\{\log \frac{\Gamma(\frac{\nu+1}{2})}{\sigma\Gamma(\frac{\nu}{2})\Gamma(\frac{1}{2})\nu^{\frac{1}{2}}} - \frac{\nu+1}{2}\log[1 + \frac{1}{\nu+\sigma^2}(x_j - \mu_i)^2]\}$$
$$+n_1 \log p(g_1) + n_2 \log(1 - p(g_2)) \tag{11}$$

where $z_{ji} = 1$ if $x_j \in g_i$ and zero otherwise. The maximum likelihood estimates can be found using the EM algorithm in the appendix B for class specific component densities with equal variances using $k_1 = k_2 = 1$. The maximum likelihood estimates of the class priors are given by the proportions of the classes in the training set, so with $n_1 = n_2$ we will estimate equal class priors and find the class boundary at the average of the estimated means.

The conditional loglikelihood in the case of class-conditional univariate Student-t distributions with equal variances is given below,

$$l_{cond} = \sum_{j=1}^{n} \delta_j \log \beta_j - \sum_{j=1}^{n} \log(1 + \beta_j) \tag{12}$$

$$\delta_j = \begin{cases} 0 & \text{if } \boldsymbol{x}_j \in g_1 \\ 1 & \text{if } \boldsymbol{x}_j \in g_2 \end{cases}, \qquad \beta_j = \frac{1 - p(g_1)}{p(g_1)} \left[\frac{1 + \frac{1}{\nu\sigma^2}(x_j - \mu_2)^2}{1 + \frac{1}{\nu\sigma^2}(x_j - \mu_1)^2}\right]^{-\frac{\nu+1}{2}}$$

For this situation we will in the following analyses estimate the class priors separately, $\hat{p}(g_i) = n_i/n$, using the maximum likelihood estimate, and then maximize the conditional loglikelihood (12) given these estimates of the class priors.

4.1.2 Joint versus Conditional Likelihood, Simulation Results

For illustration we will for a while concentrate on a subclass of the MC^2 models; one univariate Gaussian or Student-t class specific component distribution with equal variances for the classes.

We will try to draw comparative conclusions about the joint and the conditional likelihood procedure on the basis of a simulation study. The simulation study is based on data sets with two classes from the class-conditional univariate Gaussian or Student-t distribution. Since we only have two classes maximizing the conditional likelihood is identical to minimizing the the Kullback-Leibler divergence (6). We only consider equal prior class probabilities.

Small Data Set of Class-Conditional Gaussians with Equal Variances We will start by discussing the asymptotic behaviour of the joint and the conditional likelihood for class-conditional Gaussians distributions. In the case of using the MC^2 model with one univariate Gaussian class-specific component distribution with equal matrices of covariance for each class the comparison of the joint and conditional likelihood procedure is the same as comparing Fisher's linear discriminant analysis with logistic discrimination.

In Efron (1975) the joint and conditional likelihood procedures are compared for the situation of two classes with Gaussian class-conditional distributions. Let the Mahalanobis distance between the class centres be denoted, $\Delta = [(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)]^{\frac{1}{2}} = \frac{1}{\sigma}(\mu_1 - \mu_2)$. Tables for the asymptotic relative efficiency (ARE) of the error rate of the conditional with respect to the joint likelihood procedure are presented in Efron (1975) for different values for the class priors and different values of the Mahalanobis distance, Δ. In the case of equal class priors the ARE is close to one for small values of Δ ($\Delta \leq 1.5$), decreases slowly up to $\Delta = 2$ and then falls off sharply. The following numbers are taken from Efron (1975),

Δ	1	2	3	4
ARE	1.000	0.899	0.641	0.343

Fisher's linear discriminant analysis and the logistic discrimination are also compared in section 6.4.4 of Seber (1984) and section 8.5 of McLachlan (1992).

These results are asymptotical, and we are often far away the asymptotic situation. We will here study the small sample performance of the four likelihood procedures, the joint Gaussian likelihood (9), the conditional Gaussian likelihood (10), the joint Student-t likelihood (11) and the conditional Student-t likelihood (12), for training samples from class-conditional univariate Gaussian distributions with equal variances.

The training set consists of $n_1 = 10$ independent observation drawn from $X|g_1 \sim N(-1, \sigma^2)$ and $n_2 = 10$ independent observation from $X|g_2 \sim N(1, \sigma^2)$. Three different values for σ^2 are chosen, $\sigma^2 \in \{4, 1, \frac{4}{9}\}$. These parameter choices give Mahalanobis distances of $\Delta \in \{1, 2, 3\}$. For all choices, the theoretical class boundary is at $X_B = 0.0$. A total of $N = 1000$ training sets were created for each choice of the variance, σ^2. For each training set the class boundary is estimated from the above mentioned likelihoods (9)–(12). All the conditional iterative procedures used the true parameter values as the initial estimates and the joint Student-t likelihood EM algorithm used the sample mean as initial estimate of location. Four degrees of freedom are used for the Student-t distributions.

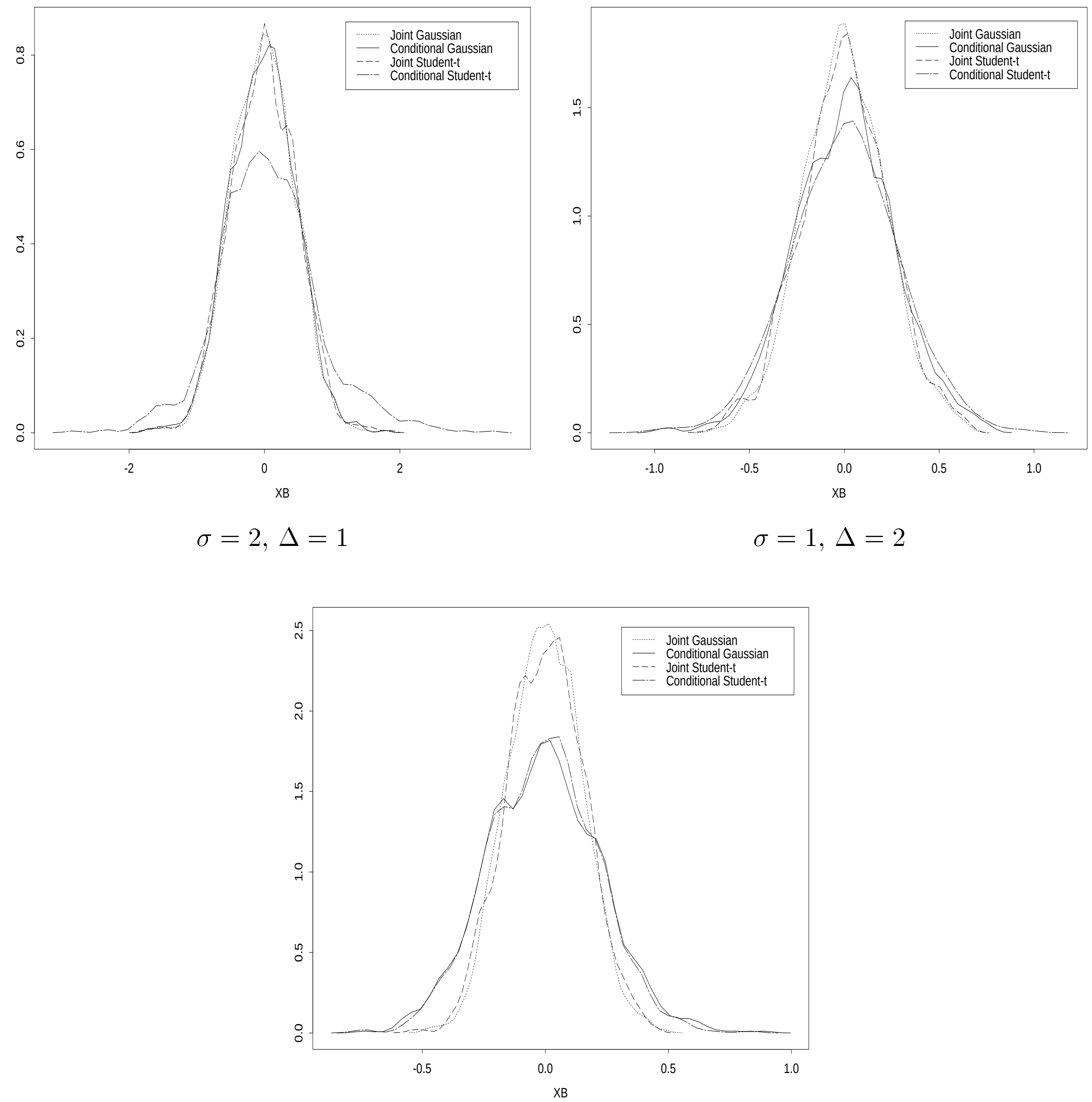
The mean and standard-deviation of the estimated class boundaries are given in table 2 and the density estimates for the estimated class boundaries are found in figure 3.

Summary Statistics for Estimated Class Boundaries for Class-Conditional Gaussian Data						
Likelihood	$\Delta = 1$		$\Delta = 2$		$\Delta = 3$	
Method	mean	std.	mean	std.	mean	std.
Joint Gaussian (9)	-0.0210	0.465	0.00114	0.222	-0.00101	0.150
Conditional Gaussian (10)	-0.0211	0.480	0.00245	0.266	-0.00727	0.228
Joint Student-t (11)	-0.0137	0.489	0.00145	0.234	0.00038	0.156
Conditional Student-t (12)	0.0565	0.749	0.00514	0.284	-0.01000	0.218

Table 2: The mean and standard deviation of 1000 estimated class boundaries for class-conditional univariate Gaussian data with two classes (with equal variances) of 10 observations each. The results are for three values for the Mahalanobis distance between the class means, Δ, and using the four likelihoods of section 4.1.1. The theoretical class boundary is at $X_B = 0.0$.

The joint Gaussian and the joint Student-t based likelihoods give nearly identical distributed estimated class boundaries for all choice of Δ. There seem to be no loss involved by using the

Density Estimates for the Estimated Class Boundary for Class-Conditional Gaussian Data



$\sigma = 2,\ \Delta = 1$

$\sigma = 1,\ \Delta = 2$

$\sigma = \frac{2}{3},\ \Delta = 3$

Figure 3: Density estimates for the estimated class boundary, $\hat{X}_B$, for two class-conditional univariate Gaussian distributions with equal variances. The estimates are based on 100 estimated class boundaries from two classes with 10 observations each. The likelihood denoted joint Gaussian (dotted line) is found in (9), the conditional Gaussian (solid line) in (10), the joint Student-t (dashed line) in (11) and the conditional Student-t (dash-dotted line) in (12). The same amount of smoothing is used for all density estimates with the same Mahalanobis distance between the class means, Δ.

joint Student-t instead of the joint Gaussian likelihood. The joint Student-t can be seen as a robustification of the joint Gaussian based likelihood procedure and should therefore be used if there is likely to be outliers present. This is the case for the small illustrative graphical example in section 4.1.3.

Comparing the conditional Gaussian based and the conditional Student-t based likelihood procedures the results are dependent on the Mahalanobis distance Δ. For small values of Δ the distribution of the estimated class boundaries from the conditional Gaussian based likelihood procedures is narrower with lighter tails than for the conditional Student-t based likelihood procedure. This difference gets smaller as Δ increases, and for $\Delta = 3$ the two procedures give nearly identical estimated distributions for the estimated class boundary.

The results from comparing the joint Gaussian with the conditional Gaussian based likelihood procedures are consistent with the results based on the asymptotic relative efficiency (ARE) earlier in this section. For small values of the Mahalanobis distance between the class means, Δ, the two procedures give nearly identically estimated distributions for the estimated class boundary, but for larger values of Δ the joint Gaussian likelihood gives better results than the conditional Gaussian based likelihood procedure. The joint Student-t based likelihood procedure is found to give a much narrower distribution for the estimated class boundary than the conditional Student-t based likelihood procedure for all choices of Δ.

We have found that the joint Gaussian based likelihood procedure is the "winner" in this small analysis, but this is of cause not surprising since we are now, as Efron puts it, "playing ball on normal discriminant analysis' home court", Efron (1975).

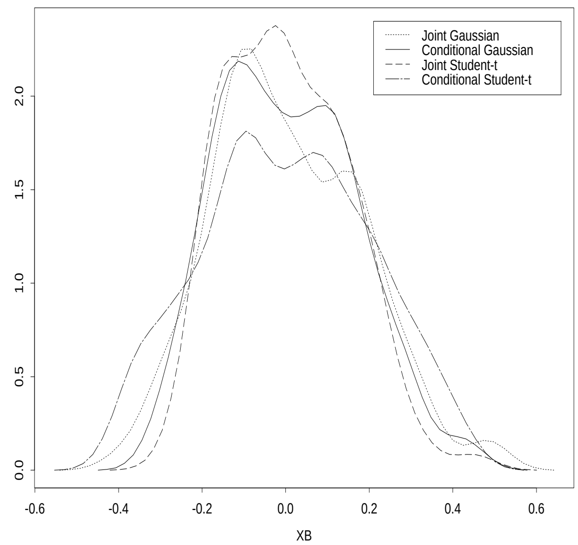
Small Data Set of Class-Conditional Student-t's with Equal Variances We will here study the performance of the four likelihood procedures (9)–(12) for training samples from class-conditional univariate Student-t distributions with equal variances and four degrees of freedom.

The training set consists of $n_1 = 100$ independent observation drawn from $X|g_1 \sim T(\mu_1 = -1, \sigma^2, \nu = 4)$ and $n_2 = 100$ independent observation from $X|g_2 \sim T(\mu_2 = 1, \sigma^2, \nu = 4)$ for three choices for σ^2; $\sigma^2 \in \{4, 1, \frac{4}{9}\}$. Since the degrees of freedom in the Student-t distributions are four, $\nu = 4$, these parameter choices give Mahalanobis distances of $\Delta \in \{\frac{1}{\sqrt{2}}, \frac{2}{\sqrt{2}}, \frac{3}{\sqrt{2}}\}$. Here $N = 100$ training sets were created for each choice of the variance, σ^2. For each training set the class boundary is estimated from the likelihoods (9)–(12) using four degrees of freedom for likelihoods based on the class-conditional Student-t distribution. As in the previous analysis, all iterative conditional likelihood routines used the true parameter values as initial estimates and the joint Student-t likelihood EM algorithm used the sample mean as initial estimate for the class locations. Summary statistics of the estimated class boundaries are found in table 3. Density estimates for the estimated class boundaries are found in figure 4.

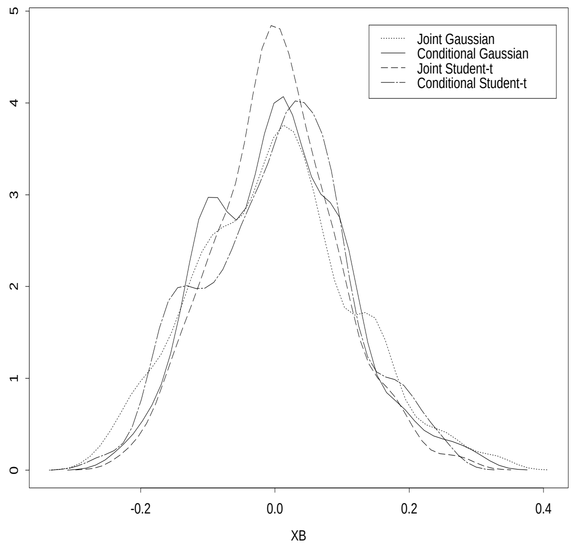
In the literature both McLachlan (1992) and Seber (1984) cite Crawley (1979) for results on comparing the linear discriminant analysis and the logistic discrimination for nonnormal data sets. When the class-conditional distributions are clearly nonnormal or their dispersion matrices are clearly unequal, Crawley found the logistic discrimination (conditional Gaussian based likelihood procedure) to be preferable to the linear discriminant analysis (joint Gaussian based likelihood procedure). This is consistent with the results found here.

Hjort (1994) considers maximum likelihood estimation in the univariate Gaussian and Student-t

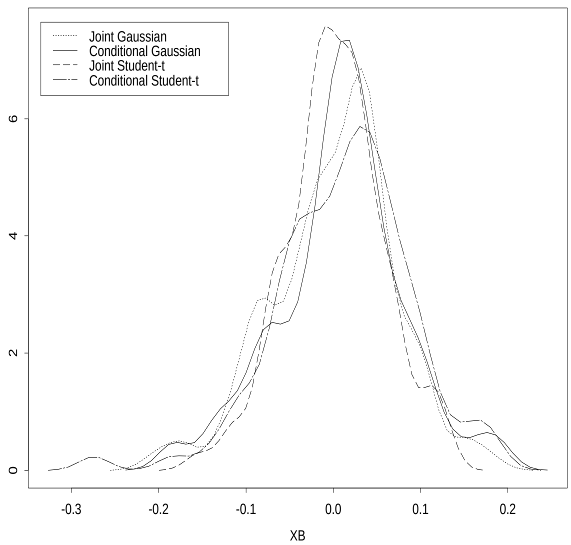
Density Estimate for the Estimated Class Boundary for Class-Conditional Student-t Data



$\sigma = 2,\ \Delta = \frac{1}{\sqrt{2}}$



$\sigma = 1,\ \Delta = \frac{2}{\sqrt{2}}$



$\sigma = \frac{2}{3},\ \Delta = \frac{3}{\sqrt{2}}$

Figure 4: Density estimates for the estimated class boundary, $\hat{X}_B$, for two class-conditional univariate Student-t distributions with equal variances. The estimates are based on 100 estimates of class boundaries from two classes of 100 observations each. The likelihood denoted joint Gaussian (dotted line) is found in (9), the conditional Gaussian (solid line) in (10), the joint Student-t (dashed line) in (11) and the conditional Student-t (dash-dotted line) in (12). The same amount of smoothing is used for all density estimates with the same Mahalanobis distance between the class means, Δ.

Summary Statistics for Estimated Class Boundaries for Class-Conditional Student-t Data									
Likelihood	$\Delta = 1/\sqrt{2}$			$\Delta = 2/\sqrt{2}$			$\Delta = 3/\sqrt{2}$		
Method	median	mean	std.	median	mean	std.	median	mean	std.
Joint Gaussian (9)	-0.00264	0.00555	0.179	0.00240	0.00059	0.116	0.00928	0.00132	0.0704
Conditional Gaussian (10)	-0.00579	0.00501	0.162	0.00139	0.00402	0.102	0.01340	0.00815	0.0721
Joint Student-t (11)	-0.01490	0.00291	0.144	0.00152	0.00217	0.090	0.00111	0.00234	0.0552
Conditional Student-t (12)	0.00466	0.00260	0.200	0.01580	0.00316	0.105	0.01990	0.01250	0.0760

Table 3: The median, mean and standard deviation of 100 estimated class boundaries for class-conditional univariate Student-t data with two classes of 100 observations each (equal variances and $\nu = 4$). The results are for three values for the Mahalanobis distance between the class means, Δ, and for the likelihoods of section 4.1.1. The theoretical boundary is at $X_B = 0.0$.

distribution. Suppose that a data set is drawn from a Student-t distribution with ν degrees of freedom. The Gaussian model corresponds to $\nu = \infty$, but will still provide more precise estimates of the location and scale parameters when $\nu \geq 1.458\sqrt{n}$, where n is the sample size (under the assumption that the degrees of freedom also has to be estimated). In our case $n = 100$, so that the Gaussian joint estimators are preferable to the Student-t estimators when $\nu \geq 14.58$. The joint Student-t based likelihood procedure should therefore in theory give the best results for the data situation analysed here. This is also found to be the case judging from the distribution of the estimated class boundaries. The estimated distributions are unimodal with low bias and low variance. Next to the joint Student-t based likelihood procedure the conditional Gaussian based likelihood procedure gives good results. The estimated distributions are slightly bimodal, but have comparable tails to those found using the joint Student-t based likelihood procedure. The conditional Gaussian based likelihood procedure gives estimated class boundary distributions that have less bias and variance than the joint Gaussian based likelihood procedure. Based on the density estimates shown in figure 4 a rough ranking of the likelihood procedures are; joint Student-t, conditional Gaussian, joint Gaussian and conditional Student-t. The important result in this case is that the conditional Gaussian likelihood procedures is better than the joint Gaussian likelihood procedure.

4.1.3 A Small Illustrative Graphical Example

To compare the five different optimization criteria (3)-(7) for the two choices of component densities, Gaussian and Student-t, a small synthetic example is analysed. This example is also analysed in Langaas (1995b) for Gaussian component densities where more details are provided.

A training set is created by drawing ten observations from each of two univariate Gaussian distributions with equal and known variances.

$$g_1: \quad x_{1,1}, x_{1,2}, ..., x_{1,10} \sim N(\mu_1 = -2, 2^2) \ \textit{and} \ g_2: \quad x_{2,1}, x_{2,2}, ..., x_{2,10} \sim N(\mu_2 = 2, 2^2)$$

We also assume known equal priors, $p(g_1) = p(g_2) = \frac{1}{2}$, so only two parameters are to be estimated, $\mathbf{\Theta} = (\mu_1, \mu_2)$. Using MAP classification the theoretical class boundary is at $X_B = 0.0$. The parameters, $\mathbf{\Theta} = (\mu_1, \mu_2)$, are estimated from the training set on the basis of

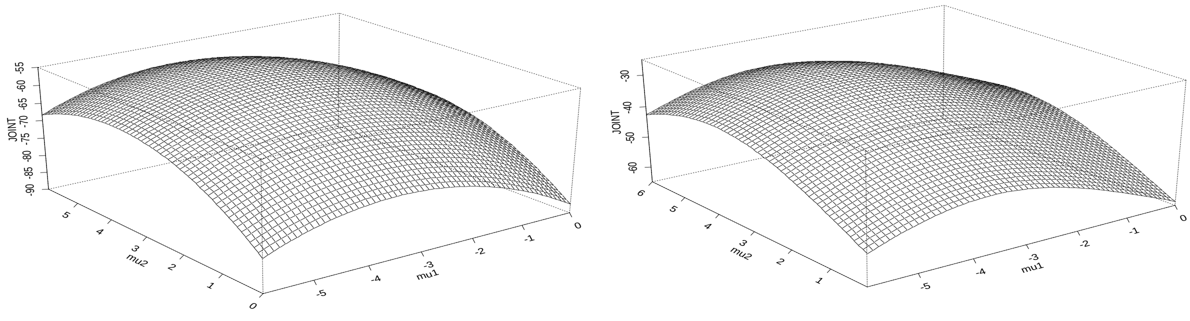
Small Illustrative Graphical Example							
Criterion	Density	value	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{X}_B$	APER	AER
JOINT	Gaussian		-2.13	2.57	0.22	10 %	16.0 %
	Student-t		-2.00	2.34	0.17	10 %	16.0 %
COND	Gaussian	-3.7	-3.87	4.45	0.29	10 %	16.1 %
	Student-t	-3.8	-3.68	4.26	0.29	10 %	16.1 %
QUAD	Gaussian	2.5	-2.91	3.37	0.23	10 %	16.0 %
	Student-t	2.5	-4.61	5.08	0.24	10 %	16.0 %
MISC	Gaussian	1	-2.29	2.21	-0.038	5 %	15.9 %
	Student-t	1	-3.50	3.19	-0.16	5 %	15.9 %

Table 4: Parameter estimates $\hat{\mathbf{\Theta}} = (\hat{\mu}_1, \hat{\mu}_2)$, the estimated class boundary $\hat{X}_B$, apparent error rate (APER) and actual error rate (AER) for the small illustrative graphical example using a data set of class-conditional univariate Gaussian densities with two classes with known and equal variances. The column "value" gives the value for the criterion functions at the parameter estimates. The criterion functions are the "JOINT" joint likelihood of (9), the "COND" conditional likelihood (10), the "QUAD" quadratic loss of (5) and the "MISC" misclassification criterion (7). The parameter estimates and the class boundary for the MISC criterion is not unique.

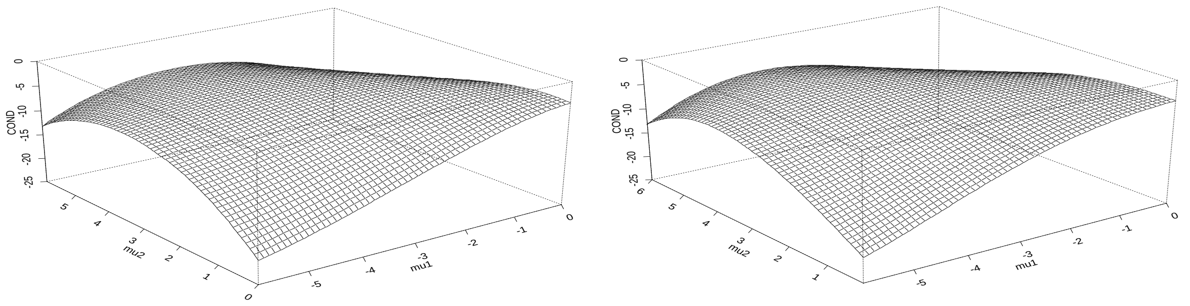
the MC^2 model with class specific component densities with $K = (1, 1)$ cluster architecture. From the previous section we have seen that the class boundary is the average of the two estimated means both when Student-t and Gaussian component densities with equal variances are used.

The four optimization criteria; joint likelihood (3), conditional likelihood (4), quadratic cost (5) and misclassification rate (7), are considered. The Kullback-Leibler criterion (6) equals the conditional likelihood criterion because we have only two classes. The optimization is done both as a grid search for $\mu_1 \in \{-5.9, -5.8, ..., 0.0\}$ and $\mu_2 = \{0.1, 0.2, ..., 6.0\}$ and also using implemented software. The optimization criteria are plotted as functions of the parameters, $\mathbf{\Theta}$, in figure 5 and as functions of the class boundary, $X_B = \frac{1}{2}(\mu_1 + \mu_2)$, in figure 6. The estimated parameters and class boundaries are found in table 4 together with the error rate on the training set (denoted APER) and the actual error rate (denoted AER). For more details on the AER in this situation see section 4.4 of Langaas (1995b). For each optimization criterion the two choices of component densities give very different parameter estimates, but give nearly the same class boundaries and the same APER and AER. In figure 7 we see the estimated means $\hat{\mu}_1$ and $\hat{\mu}_2$ together with the estimated class boundary $\hat{X}_B$. These estimates are found using implemented software.

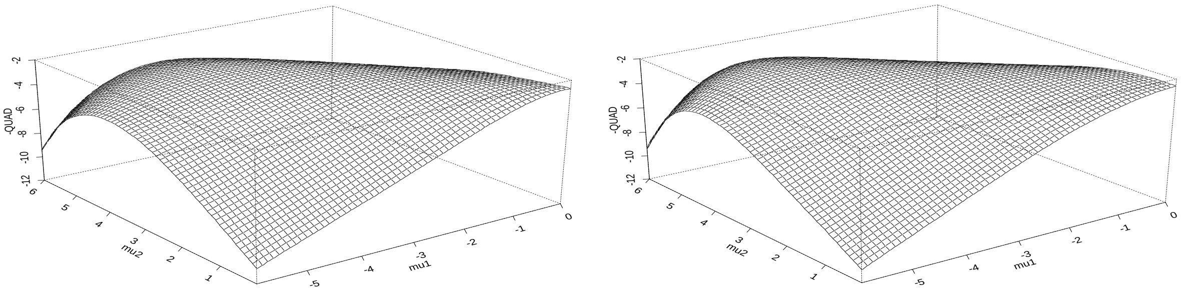
Maximizing the joint likelihood criterion the parameters are chosen solely to model the joint density of the observation. The data set analysed here is small and each class contains one outlier. For this realisation the results show that the use of Student-t component densities are more robust with respect to the estimation of the means than the Gaussian component densities. For the misclassification criterion the values for the means are not of importance, only the class boundary matters. Based on the misclassification criterion the optimal class boundary is all values in [-0.170, -0.023]. All choices for the means giving the same class



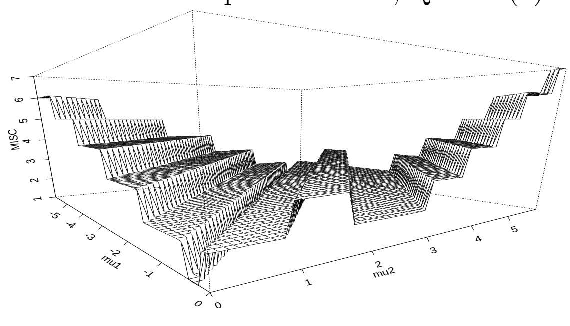
Maximize the joint loglikelihood, JOINT (3)



Maximize the conditional likelihood of correct classification, COND (4)

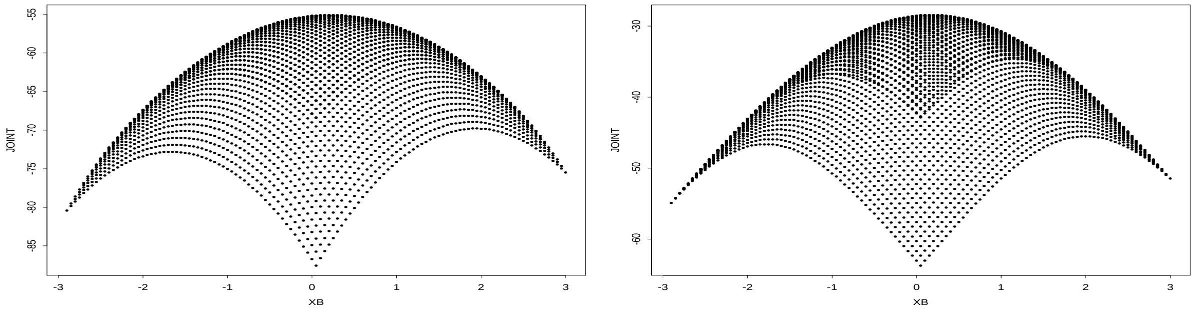


Minimize the quadratic loss, QUAD (5)

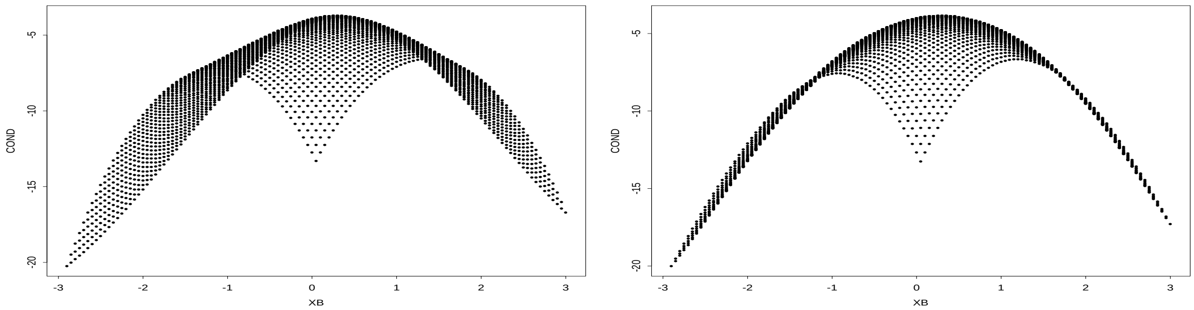


Minimize the misclassification rate, MISC (7)

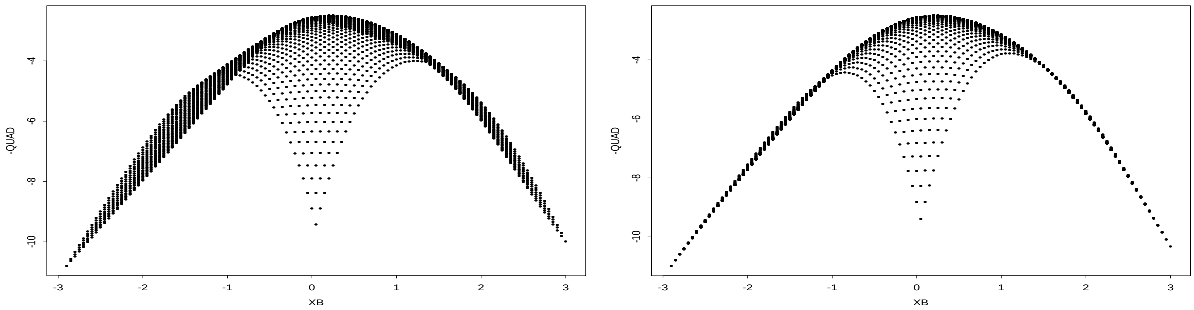
Figure 5: Plots of the criterion functions on "the small illustrative graphical example" as functions of the means, μ_1 and μ_2. The left column is for Gaussian class-conditional densities and the right for Student-t class-conditional densities with $\nu = 4$. The scale on the plot of the joint likelihood for the Gaussian and the Student-t densities are not comparable.



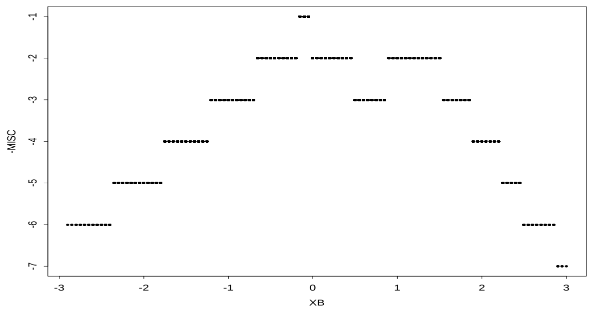
Maximize the joint loglikelihood JOINT (3)



Maximize the conditional likelihood of correct classification, COND (4)



Minimize the quadratic loss, QUAD (5)



Minimize the misclassification rate, MISC (7)

Figure 6: Plots of the criteria functions on "the small illustrative graphical example" as functions of the class boundary, X_B. The left column is for Gaussian class-conditional densities and the right for Student-t class-conditional densities with $\nu = 4$. These plots are the projection of the plots in figure 5 to the mean of the class means. Also here the scale on the plot of the joint likelihood for the Gaussian and the Student-t component densities are not comparable.

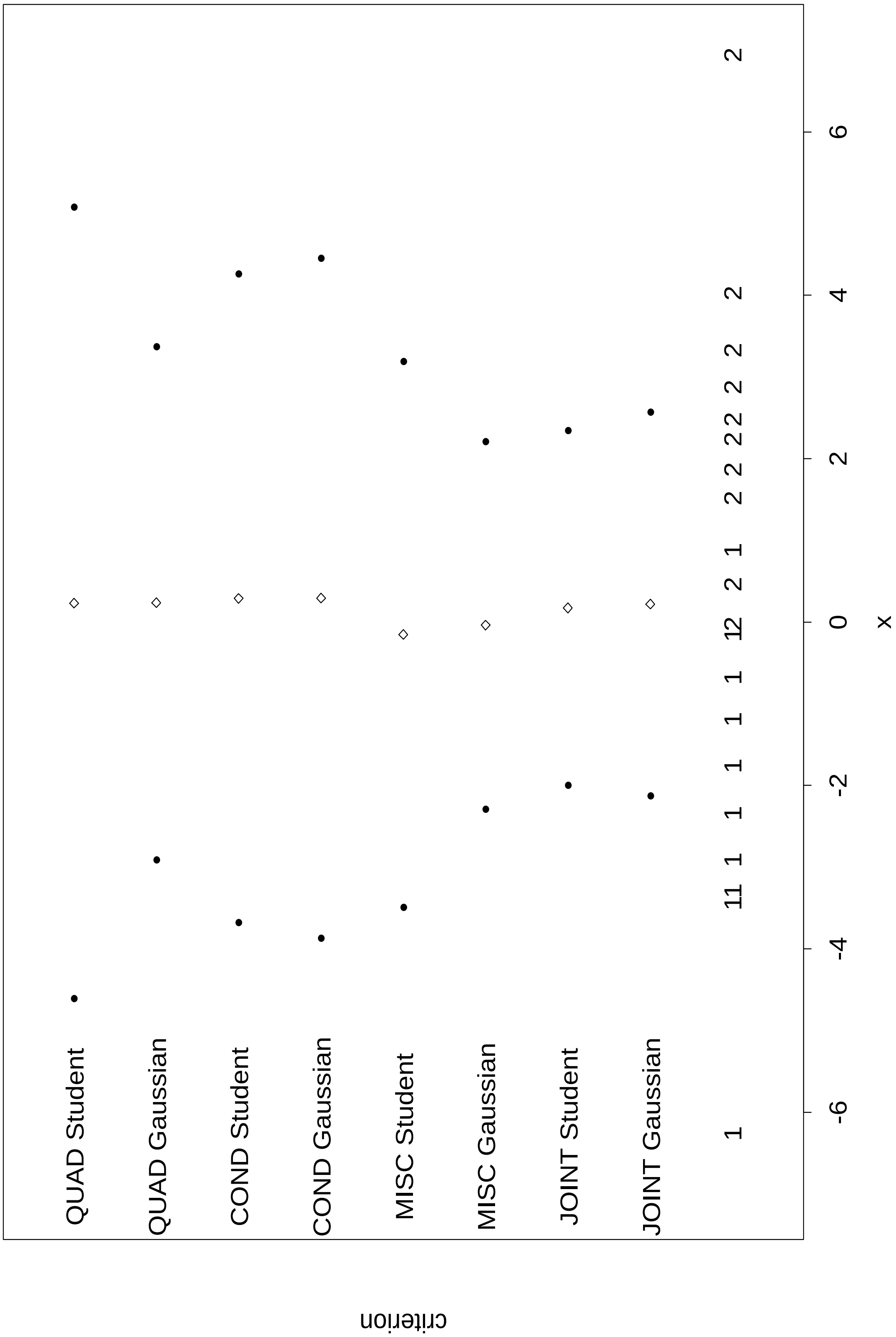


Figure 7: Estimated means and class boundaries for "the small illustrative graphical example". The parameter estimates are given by black dots and the class boundaries by open diamonds. The lower line with numbers 1 and 2 are the observations in the training set. For more details see the text.

boundary are equally attractive. For the conditional likelihood criterion and the quadratic loss criterion not only the correct classification of the training set, but also the value for the posterior probability of the correct class of each observation is taken into account. Since the posterior probabilities are functions of the parameters, this means that some choices for μ_1 and μ_2 are more desirable than others even though they give the same estimated class boundary.

4.2 Regularization

The term regularization can be interpreted as fitting a model subject to constraints put on the solution. These constrains may contain prior information about the solution, or may be based on ad hoc reasoning, see Wabha (1994).

In the MC^2 framework parameter estimation is done by optimizing a criterion function. Regularization can then be performed by imposing conditions on the parameter estimates (the solution of the optimization routine), by penalizing the optimization criterion or by stopping the optimization early (not choosing the global optimum). Penalization consists in adding a penalty to the optimization criterion. We will here consider parametric penalties, see Friedman (1994).

Let $\mathcal{J}(\mathbf{\Theta}|\mathcal{T})$ be the original optimization criterion, $P(\mathbf{\Theta})$ the penalization term and $C(\mathbf{\Theta}|\mathcal{T})$ the penalized criterion function,

$$C(\mathbf{\Theta}|\mathcal{T}) = \mathcal{J}(\mathbf{\Theta}|\mathcal{T}) + \lambda P(\mathbf{\Theta})$$

The parameter λ is called the strength, smoothing or regularization parameter. Let us assume that the criterion $\mathcal{J}(\mathbf{\Theta}|\mathcal{T})$ is to be *minimized.* The penalized task is to minimize the penalized criterion with respect to the parameters in $\mathbf{\Theta}$. For the sake of simplicity let the parameter set, $\mathbf{\Theta}$, contain R parameters; $\mathbf{\Theta} = \{\theta_r\}_{r=1}^{R}$.

A general type of penalty is the weighted bridge penalty, Friedman (1994).

$$P(\mathbf{\Theta}) = \sum_{r=1}^{R} v_r |\theta_r - \theta_r^0|^q \tag{13}$$

Here the parameter q defines the functional form of the penalty, v_r is a weighting term and θ_r^0 is some prespecified point in parameter space. As the parameter q approaches zero this will result in a subset selection. We will here consider the choice $q = 2$, which is called a ridge penalty. The penalty is then simply the sum of squared Euclidean distances from the parameter θ_r to the prespecificed point θ_r^0. In MC^2 models fitting consists in estimating the parameters $\mathbf{\Theta}$. These parameters are the component density location and scale parameters and the class and component weights. Let $\{\mu_r\}_{r=1}^{R_\mu}$ denote the location parameters and $\{\sigma_r^2\}_{r=1}^{R_\sigma}$ the diagonal elements of the scale matrices (variances and scaled variances) for a chosen MC^2 model.

Scale Penalty

$$P(\mathbf{\Theta}) = \sum_{r=1}^{R_\sigma} v_r \sigma_r^2$$

This penalty puts a restriction on large values for the variances and thereby favours small values. Large variances decrease the interpretability of the location parameters (component location). If a large regularization parameter λ is chosen the variances will approach zero and result in a nearest codeword type classification.

Conservation Penalty

$$P(\mathbf{\Theta}) = \sum_{r=1}^{R_\sigma} v_r(\sigma_r^2 - (\sigma_r^2)^0)^2 + \sum_{r=1}^{R_\mu} t_r(\mu_r - \mu_r^0)^2$$

The conservation penalty uses different weights, v_r and t_r, for the location and the scale parameters. The penalty restricts the location and scale parameters to be close to some prespecificed point in parameter space. It can be used to combine the use of two optimization criteria. For example, estimate the parameters, $\mathbf{\Theta}$, using the joint likelihood criterion and then use these estimates of locations and scales as prespecificed points, μ_r^0 and $(\sigma_r^2)^0$, in a conservation penalty when maximizing the conditional likelihood criterion. If a large regularization parameter, λ, is chosen, the locations and scales are fixed at the prespecificed values and only the class and component weights are reestimated.

Early Stopping Penalty Early stopping is a form of regularization that is very popular in the neural network community. This is usually seen as an ad hoc procedure, but Friedman (1994) defines a parametric penalty for early stopping,

$$P(\mathbf{\Theta}) = \#[iterations].$$

This penalty depends on the starting point for the optimization and the path through parameter space taken by the optimization routine. If the convergence to a (local) optimum is monotone, this can be viewed as having a cost associated with each iteration step. For more details see Friedman (1994).

The scale and conservation penalties are implemented for the optimization criteria (4)-(7). Early stopping can be performed for all criteria (3)-(7). Empirical results from using the scale penalty are presented in table 5 using the data set of Duchenne muscular dystrophy (DMD). The DMD data set is presented and analysed in section 4.3.

4.3 The Duchenne Muscular Dystrophy Carrier Data Set

We will end the topic of parameter estimation and method evaluation by comparing the performance of the MC^2 methods with other methods for discrimination and classification on a real data set. Details on the implementation of the MC^2 routines are found in appendix A.2.

The Duchenne Muscular Dystrophy Carrier (DMD) data set is presented in Andrews and Herzberg (1985), chapter 38. The purpose of collecting the data was to develop a procedure for informing a woman of the likelihood that she is a carrier of the genetically transmitted disease Duchenne muscular dystrophy. The data set used here consists of 194 observations of six explanatory variables; age, season (month) and the level of four different serum enzymes or proteins. The data set is used for classification with two classes, non-carriers and carriers.

Scale Penalties Used on the Duchenne Muscular Dystrophy Carrier Data Set Using Separate Training and Test Sets												
scale	Error rates for training set (%)						Error rates for test set (%)					
penalty	Total		Non-Carriers		Carriers		Total		Non-Carriers		Carriers	
λ	error	std	error	std	error	std	error	std	error	std	error	std
0.0	3.09	1.76	0.00	0.00	8.11	4.49	8.25	2.79	4.48	2.53	16.67	6.80
0.001	6.19	2.45	1.67	1.65	13.51	5.62	7.22	2.63	1.49	1.48	20.00	7.30
0.002	4.12	2.02	0.00	0.00	10.81	5.10	5.15	2.25	1.49	1.48	13.33	6.21
0.0025	5.15	2.25	0.00	0.00	13.51	5.62	11.34	3.22	4.48	2.53	26.67	8.07
0.003	7.22	2.63	1.67	1.65	16.22	6.06	12.37	3.34	5.97	2.89	26.67	8.07
0.005	8.25	2.79	1.67	1.65	18.92	6.44	15.46	3.67	10.45	3.74	26.67	8.07
0.01	7.22	2.63	0.00	0.00	18.92	6.44	13.40	3.46	5.97	2.89	30.00	8.37
0.015	18.56	3.95	0.00	0.00	48.65	8.22	21.65	4.18	2.99	2.08	63.33	8.80

Table 5: The effect of using scale penalties on the Duchenne Muscular Dystrophy (DMD) Carrier data set using separate training and test sets for maximizing the conditional likelihood (4) for $K = 3$ common Gaussian component densities with diagonal component specific covariance matrices. Equal weight is put on all the scale parameters, $v_r = 1/R_\sigma$. The column denoted "error" give the error rate and "std" the approximate standard deviation of the error rate (assuming the number of errors to be binomially distributed for the total data set and in each class). The columns "Total" denotes the total training or test set, "Non-Carriers" the women that are not carriers of DMD and "Carriers" the DMD carriers in the training set and in the test set. The training set is of size 94 and contains 37 DMD carriers and the test set is of size 94 and contains 30 DMD carriers. Pairwise comparisons have a 95 % confidence interval of half-length about 5 %.

The stepwise component architecture selection routine is at the current point in time only implemented for using a validation set, and not for cross-validation. To be able to explore the effects of the component architecture selection routine the DMD data set is analysed under two different settings; separate training and set sets and two layers of cross-validation.

Separate Training and Test Sets The first setting is using separate training and test sets. The total data set is randomly divided into a training set of size 97 and a test set of size 97. The number of observations from Duchenne muscular dystrophy carriers in the training and test set are 37 (38 %) and 30 (31 %), respectively.

The data set is analysed using different methods for discrimination and classification from statistics and neural networks in addition to the MC^2 models. The results from these analyses are found in table 6. Details on performing the analyses are found in appendix A.1. The lowest error rate on both the training set and the test set at the same time, is found for the multilayer perceptron trained using entropy optimization criterion and sigmoid output activation function. For the MC^2 routines the lowest error rate is found for four common Gaussian component densities with cluster specific variances (diagonal matrices of covariance) maximizing the conditional likelihood. Good results are also found for optimizing the joint likelihood. The quoted results are from using the stepwise component architecture selection routine of section 3.2.2 for class specific components and a grid search for common components (2-8 clusters). Both scale and conservation penalties are used.

This use of only one training set and one test set is biasing the comparison of the methods because the test set is used to select the best model choices within each method. This has raised the need for the second setting.

Two Layers of 10-Fold Cross-validation A first layer of 10-fold cross-validation, called the *validation-layer*, is used for model selection (within each method), and the second layer of 10-fold cross-validation, called the *test-layer*, is used to evaluate and compare the performance of the different methods.

The data set is divided randomly to produce the two 10-fold cross-validation layers. This means that in each layer 10 sets for training with associated sets for testing are produced. The validation-layer had 5 to 9 DMD carriers (28-53 %) in the testing set and the test-layer had 4 to 9 DMD carriers (22-53 %) in the testing set.

Since the use of two cross-validation layers are much more time consuming that using separate training and test sets, the range of models investigated for each method is smaller than the range used for the separate training and test set. In the validation-layer grid search is used for all the MC^2 methods. No fine-tuning using conservation penalties is done, here only moderate scale penalties are used for all the conditional likelihood methods. See Appendix A for more details on performing the analysis and the model selection.

The results on the test-layer are found in table 7. In addition to presenting the (raw) error rates also smoothing the error rate (also called risk averaging) is tried. The smoothed error rate estimator is presented in section 10.5 of McLachlan (1992), and defines a smoothed error

Duchenne Muscular Dystrophy Carrier Data Set Using Separate Training and Test Sets						
	Error rates for the test set (%)					
Method	Total		Non-Carriers		Carriers	
	error	std	error	std	error	std
Linear Discriminant Analysis	13.40	3.46	10.45	3.74	20.00	7.30
Quadratic Discriminant Analysis	15.46	3.67	14.93	4.35	16.67	6.80
1-Nearest Neighbour Classification	13.40	3.46	13.43	4.17	13.33	6.21
9-Nearest Neighbour Classification	10.31	3.09	2.99	2.08	26.67	8.07
Classification tree (default)	17.53	3.86	11.94	3.96	30.00	8.37
Projection Pursuit Regression (1)	8.25	2.79	4.48	2.53	16.67	6.80
Generalized Additive Model (logit)	7.22	2.63	5.97	2.89	10.00	5.48
Flexible Discriminant Analysis with MARS (1)	10.31	3.09	4.48	2.53	23.33	7.72
ditto with MARS (2)	15.46	3.67	11.94	3.96	23.33	7.72
ditto with BRUTO	8.25	2.79	4.48	2.53	16.67	6.80
Multilayer Perceptron 6-7-1 (ent)	8.25	2.79	7.46	3.21	10.00	5.48
Multilayer Perceptron 6-2-1 (lin)	8.25	2.79	5.97	2.89	13.33	6.21
Multilayer Perceptron 6-7-1 (sig)	10.31	3.09	10.45	3.74	10.00	5.48
Learning Vector Quantizer (3 codewords)	11.34	3.22	2.99	2.08	30.00	8.37
MC^2 JOINT Gaussian $N(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}_i))$ K=(1,2)	7.22	2.63	2.99	2.08	16.67	6.80
MC^2 JOINT Student-t $T(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$ K=(1,1)	9.28	2.95	7.46	3.21	13.33	6.21
MC^2 JOINT Gaussian $N(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}_l))$ K=4	7.22	2.63	2.99	2.08	16.67	6.80
MC^2 JOINT Student-t $T(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}_l))$ K=3	9.28	2.95	4.48	2.53	20.00	7.30
MC^2 COND Gaussian $N(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$ K=(2,4)	11.34	3.22	10.45	3.74	13.33	6.21
MC^2 COND Student-t $T(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$ K=(1,3)	9.28	2.95	7.46	3.21	13.33	6.21
MC^2 COND Gaussian $N(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}_l))$ K=3	5.15	2.25	1.49	1.48	13.33	6.21
MC^2 COND Student-t $T(\boldsymbol{\mu}_l, diag(\boldsymbol{\Sigma}_l))$ K=2	10.31	3.09	7.46	3.21	16.67	6.80

Table 6: Error rates (%) for the test set for the Duchenne muscular dystrophy data set using separate training and test set. The test set is of size 97 and contains 30 DMD carriers. The columns denoted “error” give the error rate (raw) on the test set for “Total” (the total test set), “Non-carriers” (women that are not carriers of DMD) and “Carriers” (DMD carriers). The columns marked “std” give approximate standard deviations for the errors by assuming a binomial distribution for the number of errors for the total test set and for each class. Comparing any two methods based on the error on the total test set using the standard deviation concludes with significant difference only for comparing the classification tree and the MC^2 COND Gaussian with $K = 3$ common components on a 5% significance level (the methods with the highest and lowest error rate). Pairwise comparisons (of the total error rate) have a 95 % confidence interval of half-length about 5%. Details on the model selection are found in appendix A.1.

rate based on the smallest posterior class probability for each observation,

$$A_{smooth} = \frac{1}{n_{test}} \sum_{j=1}^{n_{test}} \min_{i=1,2} \hat{p}(g_i|\boldsymbol{x}_j)$$

where n_{test} is the number of observation in the test set. The mean and its standard deviation over the 10 cross-validation runs for this smoothed error rate for the test-layer are quoted in table 7 together with the mean and its standard deviation for the raw error rates.

5 Conclusions and Comments

The basis for the mixture codebook classification (MC^2) lies in the construction of a mixture codebook, $\mathcal{M} = \{\boldsymbol{\Theta}, f, K\}$, as outlined in "Mixture Codebook Classification, Part 1: Method Outline" by Langaas (1995b). In the present study additional features have been added to the MC^2 framework. These include the use of multivariate Student-t distributions in the class-conditional mixture models, a stepwise routine for architecture selection and performing regularization using parametric penalties. Hopefully, also insight has been provided into the comparative performance of the joint (3) and the conditional likelihood (4).

Some general conclusions can be drawn about MC^2. MC^2 can produce flexible and nonlinear class boundaries, but also robust linear boundaries when needed. It is based on a parametric model, but the use of class-conditional mixture models have made the assumptions on the class-conditional distributions flexible. MC^2 has the possibility of handling outliers and wrongly classified (labelled) observations. If there is a possibility that a data set contains outliers or wrongly classified observations it is of importance that robust estimation is performed. In section 4.1.2 and 4.1.3 we have seen that for the simple case of maximizing the joint likelihood with one component density for each class the use of Student-t distributions (with $\nu = 4$ degrees of freedom) instead of Gaussian distributions, has given good results and improved error rates. The usefulness of the Student-t distribution, and mixtures thereof, is supported by the results for the full classification of the forensic glass data set in section 3.1.3 for the joint and the conditional likelihood criteria for both common and class specific component architectures.

In section 3.1 some concern was expressed with respect to using the EM algorithm for Gaussian and Student-t mixtures. Important topics were the choice of starting (initial) parameters for the algorithm, the speed of convergence and the estimated parameters that the EM algorithm converge to. In this study the EM algorithm has in separate cases diverged, that is, it has arrived at singular scale matrices, but in most cases the convergence has been to a local maximum inside the parameter space. The starting point for the EM algorithm have in this study been chosen using a K-means algorithm. The use of multiple initial estimates for the K-means algorithm has been used, but this has not lead to that the EM algorithm has arrived at different local maxima. Parameter estimates corresponding to a so called less desirable local maximum of the loglikelihood are reported to be a potential problem in Ingrassia (1992). The aim of constructing a mixture codebook is to perform classification, and not density estimation. There might exist situations where a so called less desirable local maximum of the loglikelihood is the most desirable local maximum for constructing the classifier. More work is needed on assessing the convergence and parameter estimates for the multivariate Student-t mixtures.

Duchenne Muscular Dystrophy Carrier Data Set Using Two Layers of 10-Fold Cross-validation								
	Raw Error Rates (%)						Smoothed (%)	
Method	Total		Non-Carriers		Carriers		Total	
	ave	std	ave	std	ave	std	ave	std
Linear Discriminant Analysis	11.39	2.19	3.89	1.70	25.08	7.30	10.07	0.76
Quadratic Discriminant Analysis	11.29	1.66	7.75	1.47	17.12	4.64	7.12	0.83
7-Nearest Neighbour Classification	10.82	1.62	2.02	1.39	27.48	3.74	-	-
Classification tree (default)	15.92	2.43	11.98	3.04	21.46	4.81	5.82	0.81
Projection Pursuit Regression (1)	9.76	1.62	3.46	2.08	23.13	4.83	8.63	1.05
Generalized Additive Model (logit)	9.24	1.48	2.39	1.25	22.71	3.90	8.11	0.93
Flexible DA with MARS (1)	11.29	2.13	2.05	1.50	28.73	5.78	5.24	0.73
ditto with MARS (2)	12.34	1.92	4.28	1.84	27.71	5.23	2.60	0.56
ditto with BRUTO	9.76	1.80	2.46	1.28	24.80	6.01	6.33	0.68
Multilayer Perceptron 6-1-1 (lin)	12.42	2.10	4.80	1.72	26.74	7.28	-	-
Multilayer Perceptron 6-4-1 (sig)	26.84	4.12	4.11	2.18	70.65	12.33	-	-
Multilayer Perceptron 6-6-7 (ent)	13.89	3.16	3.39	2.06	37.02	10.99	5.70	1.29
Learning Vector Quantizer (7 codewords)	16.50	1.71	0.62	0.63	46.82	4.91	-	-
MC^2 JOINT $N(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$ K=(2,4)	9.24	1.48	4.08	1.40	20.04	5.16	7.88	0.78
MC^2 JOINT $T(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$ K=(1,1)	11.34	1.69	6.19	2.03	20.63	4.73	7.18	0.97
MC^2 JOINT $N(\boldsymbol{\mu}_{l}, diag(\boldsymbol{\Sigma}_l))$ K=2	10.79	1.93	2.02	1.39	27.99	4.44	9.43	0.58
MC^2 JOINT $T(\boldsymbol{\mu}_{l}, diag(\boldsymbol{\Sigma}_l))$ K=2	9.79	1.63	3.73	1.26	20.63	4.73	8.24	0.74
MC^2 COND $N(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$ K=(1,1)	10.26	2.00	5.49	2.05	20.35	4.15	8.55	0.79
MC^2 COND $T(\boldsymbol{\mu}_{il}, diag(\boldsymbol{\Sigma}))$ K=(1,1)	9.24	1.48	3.23	1.35	21.46	4.02	8.34	0.84
MC^2 COND $N(\boldsymbol{\mu}_{l}, diag(\boldsymbol{\Sigma}))$ K=2	10.79	1.93	4.62	1.28	22.06	6.14	6.01	0.75
MC^2 COND $T(\boldsymbol{\mu}_{l}, diag(\boldsymbol{\Sigma}))$ K=2	9.76	1.42	2.88	1.18	22.3	4.09	5.89	0.92

Table 7: Error rates (%) for the Duchenne muscular dystrophy data set using two layers of cross-validation. The columns denoted "Raw Error Rates" give raw error rates and the columns denoted "Smoothed" give smoothed error rates. The columns denoted "Total" deal with both classes, "Non-carriers" with only the women that are not carriers of DMD and "Carriers" with the DMD carriers. The quoted error rates are the "ave" average and the "std" standard deviation of the average for the 10 cross-validation runs in the test-layer. The classification methods that are not producing posterior class probabilities are denoted with "-". We see that the mean smoothed error rate is below the mean raw error rate for all methods. In some cases the difference for the raw and smoothed error rates is large. This might suggest that the posterior probabilities produced by these methods are not correctly calibrated, and further studies are needed to check this. Comparisons based on the raw error on the total test set using the standard deviation conclude with no significant difference for any pairs of the methods on an approximate 5 % level, except for comparisons involving the learning vector quantizer and the multilayer perceptron (sig). The multilayer perceptron (sig) seems to have found a less desirable local maximum in the parameter space.

We will give some rough estimates of the number of iterations needed for the EM algorithm for Gaussian and Student-t mixtures. For small data sets (below 100 observations) and for moderate number of Gaussian component densities the EM algorithm converges in about 10 iterations. For larger problems 20 to 50 iterations are needed. For the Student-t component densities these numbers are approximately doubled. For the nonlinear optimization of the conditional likelihood criterion, and the other posterior based criteria, the search for a (local) optimum can be slow. The type of nonlinear optimization required for the conditional likelihood has in the RBF network literature, Xu et al. (1994), often been regarded to be hard and sequential solutions to parameter estimation are used instead. The sometimes slow convergence of the downhill simplex method for the conditional likelihood criterion must be seen together with the good results, with respect to error rates, that often are obtained.

The strength of the MC^2 framework lies in its flexibility in model and method choice. But, without guidelines for model and method selection the flexibility can be seen as a potential weakness. In this report the "best" MC^2 model with respect to component architecture and specification of the covariance matrices, is selected for eight situation. These situations are the combinations of using common or class specific components, Gaussian or Student-t component distributions, and joint or conditional likelihood optimization. A good starting point for investigating the MC^2 framework is the "vanilla MC^2"; class specific Gaussian component distributions with equal (diagonal) matrices of covariance for the classes and maximizing the joint likelihood criterion. The stepwise component architecture selection routine of section 3.2.2 can be used to find a good component architecture. The route through the MC^2 framework can then proceed by regarding the conditional likelihood optimization criterion for the chosen model (with the possibility of including penalties). The use of Student-t component distributions should then be investigated, first for joint likelihood optimization, but finally also for conditional likelihood optimization. The same route can be taken for the situation with common components for the classes (using a grid search for component architecture selection). More work can be put into making the MC^2 more easily available to the potential user.

We will close by commenting on the graphical interpretability of the MC^2. In some situations it might be of interest to produce low dimensional plots of the data using transformations of the variables that provide good separation of the classes and the components The MDA of Hastie and Tibshirani (1995) includes such a graphical feature. Fisher's linear discriminant analysis is based on finding linear combinations of the data that maximize the ratio of the between-class to the within-class variability. This can be extended to finding linear combinations that maximizes the ratio of the between-component to the within-component variability. For the MC^2 situation with common components with equal matrices of covariance, $\mathbf{\Sigma}$, this becomes,

$$\begin{aligned}\boldsymbol{B}_c &= \sum_{l=1}^{K} (\boldsymbol{\mu}_l - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_l - \bar{\boldsymbol{\mu}})^T \\ \boldsymbol{W}_c &= \mathbf{\Sigma}\end{aligned}$$

where $\boldsymbol{B}_c$ denotes the between-component variability and $\boldsymbol{W}_c$ denotes the within-component variability, $\boldsymbol{\mu}_l$ is the location of the lth component density and $\bar{\boldsymbol{\mu}}$ is the overall mean of the component locations. Proceeding as for Fisher's linear discriminant analysis the graphical linear combinations will be the scaled eigenvectors of $\boldsymbol{W}_c^{-1}\boldsymbol{B}_c$ corresponding to the non-zero eigenvalues of $\boldsymbol{W}_c^{-1}\boldsymbol{B}_c$. A similar graphical solution is possible for the class specific components with equal matrices of covariance. Investigating the powers of this linear transformation and other nonlinear transformations will be the topic of further research by the author.

Acknowledgements

I would like to thank to my supervisor Bo Lindqvist for helpful discussions and suggestions. This report was written while the author was funded by a Dr.Ing. grant from the Norwegian Research Council.

A Some Details on Implementation and Analysis

A.1 Discrimination and Classification

The analyses are done using the S language. General introductions to the S language are Becker et al. (1988), Chambers and Hastie (1992) and Venables and Ripley (1994).

The linear and quadratic discriminant analysis, kth nearest neighbour and learning vector quantizer are performed using the `classif` library of B.D. Ripley. The flexible discriminant analysis (flexible DA) is performed using the `fda` library of T. Hastie and R. Tibshirani. The multilayer perceptrons are trained with the `nnet` library of B.D. Ripley. All S-Plus libraries are available from the statistics archives at the Carnegie-Mellon University (`statlib@lib.stat.cmu.edu`).

For the projection pursuit regression, generalized additive model and multilayer perceptrons one binary response is used for the situation with two classes. For the multilayer perceptrons three combinations of output activation functions and loss functions are used; sigmoid units and quadratic loss (sig), sigmoid units and entropy loss (ent) and linear units with quadratic loss (lin). The multilayer perceptrons have one hidden layer with sigmoid activation function and skip layer connections between the input and output layer. No weight decay is used. The generalized additive models have logit link.

Details on the Model Selection for the Duchenne Muscular Dystrophy Carrier Data Set Using Separate Training and Test Sets The nearest neighbour classification is performed with the 1, 3, 5, 7 and 9 nearest neighbours. For the projection pursuit regression 1 to 5 projections are tried. The generalized additive model is fitted using second degree splines for each explanatory variable. The explanatory variables that appear linear are then substituted by linear terms. The multilayer perceptrons have 1 to 8 hidden nodes and are run 10 times for each architecture to find a good local maximum evaluated by the value of the minimization criterion. The learning vector quantizer have from 1 to 8 codewords and are run 5 times to find a good local maximum evaluated using the error rate. The MC^2 method with class specific components use the component architecture selection routine of section 3.2.2 with the test set to perform the duties of the validation set. The upper limit on the component architectures is set at K=(4,4). The MC^2 methods with common components use a grid search for 2 to 8 components. The use of penalties are done heuristically. All types of covariance structures are tried; full, diagonal and diagonal with equal variances for class specific covariance matrices and equal covariance matrices for the classes or components. Multiple starting points are used heuristically for the EM algorithm. For the simplex routines 5 starting points are used.

Details on the Model Selection for the Duchenne Muscular Dystrophy Carrier Data Set Using Two Layers of Cross-validation The model selection for the two layers of cross-validation is basically the same as for the separate training and test sets, so here only the differences are addressed. The multilayer perceptrons have 1 to 8 hidden nodes and are run 5 times for each architecture to find a good local maximum evaluated by the value of the minimization criterion. The learning vector quantizer have from 1 to 8 codewords and are run only once. All MC^2 method use grid search for the component architecture selection. For class specific components the following component architectures are considered K={(1,1),(1,2),(2,3),(2,4)} and for common components K={2,3,4}. Also here the use of penalties are done heuristically. Only diagonal covariance matrices are considered both for class specific covariance matrices and equal covariance matrices for the classes or components. Multiple starting points are used heuristically for the EM algorithm. For the simplex routines 3 starting points are used.

A.2 Details on the MC^2 Implementation and Analyses

Routines for MC^2 estimation and prediction are implemented in the S language with an C interface. The routines handles both class specific and common components and the five optimization criteria, (3)-(7). Both Gaussian and multivariate Student-t component distributions (with fixed degrees of freedom) are supported. The stepwise component architecture and grid search routines of section 3.2.2 are also implemented.

The joint likelihood (3) is maximized using the EM algorithm and the remaining four criteria (4)-(7) that are based on the posterior distributions for the classes, are optimized using the downhill simplex algorithm `amoeba` from Press et al. (1988).

The joint likelihood implementation handles covariance matrices that are full, diagonal and diagonal with equal variances (scale matrices for the Student-t case) and allows for specifying the prior probabilities of the classes. Only regularization by early stopping is provided for the optimization of the joint likelihood. [1]

The implementation of the posterior distribution based optimization criteria (4)-(7) handle diagonal covariance matrices and diagonal covariance matrices with equal variances. The elements of the scale matrices are transformed to avoid negative variances. Instead of class conditional component probabilities and prior class probabilities unscaled joint component and class weights are used (restricted to lie in [0,1]). This reparameterisation is done to avoid all sum-to-one constraints on the component and class probabilities. For the situation with class specific components the joint weights are $v_{il} = p(c_{il}, g_i) \cdot v$, where $v = \sum_{i=1}^{M} \sum_{l=1}^{k_i} v_{il}$. Outside the optimization routine the weights are scaled to produce joint component and class probabilities, $p(c_{il}, g_i)$, and then $p(g_i) = \sum_{l=1}^{k_i} p(c_{il}, g_i)$ and $p(c_{il}|g_i) = p(c_{il}, g_i)/p(g_i)$. This

[1]The case of maximizing the joint likelihood using class specific components with Gaussian component densities and equal (full) matrices of covariance is implemented in the `mda` library of T. Hastie and R. Tibshirani, Hastie and Tibshirani (1995). Their implementation include different types of explicit regularization (rank reduction and penalization of the between component variability within the classes).

means that the prior probabilities are not estimated by the proportions of the classes in the training set and can not be specified by the data analyst. Common components are treated in the same manner. For class specific components uniform joint component and class weights are also supplied. For the posterior probability based routines the scale and conservation penalties of section 4.2, are included. Early stopping is also possible.

The K-means algorithm of Hartigan and Wong (1979) (as implemented in S-Plus) is used to find a good starting point for the implemented EM algorithms. The K-means algorithm is initialized by drawing observations randomly from the training set. Routinely only one starting point for the EM algorithm is tried for each MC^2 model, but in some cases also multiple starting points are chosen. The EM algorithm is stopped when the average change in each parameter estimate (the means and covariance matrices) is less than 0.005. Let θ_r denote a general parameter, and $\theta_r^{(s)}$ the parameter estimates at the sth step of the algorithm. The stopping (convergence) criterion is as follows,

$$\frac{1}{R}\sqrt{\sum_{r=1}^{R}(\theta_r^{(s+1)} - \theta_r^{(s)})^2} \leq 0.005.$$

The criterion functions using the downhill simplex routine are started each time with $antpar + 1$ initial starting points, where $antpar$ equals the number of parameters and weights to be estimated. These starting points are perturbed versions of the parameter estimates from the EM algorithm. The downhill simplex routines are run five times from these different perturbed start points. The routine is said to have converged if the change in the criterion function used is smaller than 10^{-6}. Let the criterion function at the sth step of the routine be denoted $\mathcal{J}(\mathbf{\Theta}^{(s)}|\mathcal{T})$. The routine is then stopped (considered converged) when

$$\frac{|\mathcal{J}(\mathbf{\Theta}^{(s+1)}|\mathcal{T}) - \mathcal{J}(\mathbf{\Theta}^{(s)}|\mathcal{T})|}{\frac{1}{2}[\mathcal{J}(\mathbf{\Theta}^{(s+1)}|\mathcal{T}) + \mathcal{J}(\mathbf{\Theta}^{(s)}|\mathcal{T})]} \leq 10^{-6}.$$

The number of iterations needed for the implemented EM algorithm for Gaussian and Student-t class-conditional mixtures has in the examples studied here been below 100, and in most cases significantly smaller. The implemented downhill simplex optimization of the conditional likelihood may need up to 5000 iterations (and some times even more), but each iterations is much faster than for the EM algorithm.

B The EM algorithm for Mixtures of Multivariate Student-t Component Distributions

General references to the EM algorithm are Dempster et al. (1977) and Little and Rubin (1987). References to the EM algorithm in connection with multivariate Student-t distributions are Lange et al. (1989), Kent et al. (1994), Rubin (1983) and Arslan et al. (1993).

B.1 One Component

The training set, $\mathcal{T} = (\boldsymbol{x}_j)_{j=1}^n$, consists of independent observations from the multivariate Student-t distribution, $\boldsymbol{X}_j \sim T_d(\boldsymbol{\mu}, \mathbf{\Sigma}, \nu)$.

The philosophy behind the EM algorithm is to start with a problem with a known (or easy) solution. We start by regarding a conditional Gaussian distribution, Lange et al. (1989),

$$\boldsymbol{X}_j | t_j \sim N_d(\boldsymbol{\mu}, \boldsymbol{\Sigma}/t_j) \qquad\qquad t_j \sim \chi^2_\nu/\nu$$

$$t_j|\boldsymbol{x}_j \sim \chi^2_{\nu+d}/[\nu + (\boldsymbol{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu})].$$

We augment the observed data $(\boldsymbol{x}_j)_{j=1}^n$ by unobserved data $(t_j)_{j=1}^n$ such that marginally the observations $\boldsymbol{x}_j$ have a multivariate Student-t distribution; $\boldsymbol{X}_j \sim T_d(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$. We regard the degrees of freedom as a parameter. The parameters are denoted $\boldsymbol{\Theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$. The complete data are $(\boldsymbol{x}_j, t_j)_{j=1}^n$, yielding the complete data loglikelihood.

$$l = \log(\prod_{j=1}^n p(\boldsymbol{x}_j, t_j)) = \log(\prod_{j=1}^n p(\boldsymbol{x}_j|t_j)p(t_j))$$

$$= n[\frac{1}{2}\log|\boldsymbol{\Sigma}|^{-1} - \frac{d}{2}\log 2\pi + \log\nu - \frac{\nu}{2}\log 2 - \log\Gamma(\frac{\nu}{2}) + (\frac{\nu}{2} - 1)\log\nu]$$

$$+ \sum_{j=1}^n [(\frac{d}{2} + \frac{\nu}{2} - 1)\log t_j - \frac{t_j}{2}(\nu + (\boldsymbol{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}))]$$

The EM algorithm alternates between averaging the complete data loglikelihood, l, over the unobserved scalings, $\{t_j\}_{j=1}^n$, and maximizing the averaged loglikelihood over the parameters, $\boldsymbol{\Theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$.

The expectation step of the algorithm (E-step) calculates the expectation of the complete data loglikelihood given the observed data $\mathcal{T} = (\boldsymbol{x}_j)_{j=1}^n$ and the current estimate of the parameters, $\boldsymbol{\Theta}^{(s)}$.

$$E(t_j|\mathcal{T}, \boldsymbol{\Theta}^{(s)}) = \frac{\nu^{(s)} + d}{\nu^{(s)} + (\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)})^T (\boldsymbol{\Sigma}^{(s)})^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)})}$$

$$E(\log t_j|\mathcal{T}, \boldsymbol{\Theta}^{(s)}) = DG(\frac{\nu^{(s)} + d}{2}) - \log 2 - \log[\nu^{(s)} + (\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)})^T (\boldsymbol{\Sigma}^{(s)})^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)})]$$

where DG denotes the digamma function, $DG(x) = \frac{d\log\Gamma(x)}{dx}$. The last result follows from the fact that for the variable $w_j = [\log t_j + \log(\nu + (\boldsymbol{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu})) + \log 2]$ the conditional the distribution of $w_j|\boldsymbol{x}_j$ is a log-Gamma distribution with expected value $DG(\frac{\nu+d}{2})$.

Let us for simplicity denote $D_j^{(s)} = (\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)})^T (\boldsymbol{\Sigma}^{(s)})^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)})$. The expected complete data loglikelihood is then given below.

$$E(l|\mathcal{T}, \boldsymbol{\Theta}^{(s)}) = n[\frac{1}{2}\log|\boldsymbol{\Sigma}|^{-1} - \frac{d}{2}\log 2\pi + \log\nu - \frac{\nu}{2}\log 2 - \log\Gamma(\frac{\nu}{2}) + (\frac{\nu}{2} - 1)\log\nu]$$

$$+ n(\frac{d}{2} + \frac{\nu}{2} - 1)(DG(\frac{\nu^{(s)} + d}{2}) - \log 2) - (\frac{d}{2} + \frac{\nu}{2} - 1)\sum_{j=1}^n \log(\nu^{(s)} + D_j^{(s)})$$

$$- \frac{1}{2}\sum_{j=1}^n \frac{\nu^{(s)} + d}{\nu^{(s)} + D_j^{(s)}}[\nu + (\boldsymbol{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu})]$$

The maximization step (M-step) consists in maximizing the function $E(l|\mathcal{T}, \mathbf{\Theta}^{(s)})$ with respect to $\mathbf{\Theta}$. We first define weights.

$$w_j^{(s)} = \frac{\nu^{(s)} + d}{\nu^{(s)} + (\boldsymbol{x}_j - \boldsymbol{\mu}^{(s)})^T (\mathbf{\Sigma}^{-1})^{(s)} (\boldsymbol{x} - \boldsymbol{\mu}^{(s)})} = \frac{\nu^{(s)} + d}{\nu^{(s)} + D_j^{(s)}} \qquad for \qquad j = 1, .., n$$

This leads to the following M-step equations.

$$\begin{aligned} \boldsymbol{\mu}^{(s+1)} &= \frac{\sum_{j=1}^n w_j^{(s)} \boldsymbol{x}_j}{\sum_{j=1}^n w_j^{(s)}} \\ \mathbf{\Sigma}^{(s+1)} &= \frac{1}{n} \sum_{j=1}^n w_j^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}^{(s+1)})(\boldsymbol{x}_j - \boldsymbol{\mu}^{(s+1)})^T \end{aligned}$$

In the equation for $\mathbf{\Sigma}^{(s+1)}$ the $\boldsymbol{\mu}^{(s+1)}$ could be substituted by $\boldsymbol{\mu}^{(s)}$.

The M-step updates $\nu^{(s+1)}$ by maximizing the expected total data loglikelihood with respect to ν.

$$\nu^{(s+1)} = \underset{\nu}{\operatorname{argmax}} \{ \nu \log \frac{\nu}{2} - 2 \log \Gamma(\frac{\nu}{2}) + \nu [DG(\frac{\nu^{(s)} + d}{2}) - \log 2]$$

$$- \frac{\nu}{n} \sum_{j=1}^n \log(\nu^{(s)} + D_j^{(s)}) - \frac{\nu}{n} \sum_{j=1}^n \frac{\nu^{(s)} + d}{\nu^{(s)} + D_j^{(s)}} \}$$

The EM algorithm is schematically outlined for univariate data in appendix A.2 of Lange et al. (1989).

B.2 Class-Conditional Mixture with Class Specific Component Distributions

The training set consists of observations and encoded class memberships, $\mathcal{T} = (\boldsymbol{x}_j, \boldsymbol{z}_j)_{j=1}^n$, where $\boldsymbol{z}_j$ denotes the encoded class of observation $\boldsymbol{x}_j$. The data can be regarded as incomplete, the component membership is missing. The component membership is encoded as $\boldsymbol{æ}_j$. We proceed in a similar manner to the previous section. Suppose that $\boldsymbol{x}_j$ is a member of the group g_i and component c_{il}.

For common scale matrices for all components, we have the following.

$$\boldsymbol{X}_j | t_j, \boldsymbol{z}_j, \boldsymbol{æ}_j \sim N_d(\boldsymbol{\mu}_{il}, \mathbf{\Sigma}/t_j) \qquad t_j \sim \chi^2_\nu / \nu$$

$$t_j | \boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{æ}_j \sim \chi^2_{\nu+d} / [\nu + (\boldsymbol{x}_j - \boldsymbol{\mu}_{il})^T \mathbf{\Sigma}^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_{il})]$$

In the following the scale parameter $\mathbf{\Sigma}$ is substituted by $\mathbf{\Sigma}_i$ for the case of separate scale matrices for the classes and by $\mathbf{\Sigma}_{il}$ for separate scale matrices for the components.

The *complete* training set is $(\boldsymbol{x}_j, t_j, \boldsymbol{z}_j, \boldsymbol{æ}_j)_{j=1}^n$. We start by finding the loglikelihood for the complete training set. The parameters are denoted
$\mathbf{\Theta} = \{(\boldsymbol{\theta}_{il}, p(c_{il}|g_i), p(g_i)); i = 1, .., M; l = 1, ..., k_i\}$ where $\boldsymbol{\theta}_{il} = \{\boldsymbol{\mu}_{il}, \mathbf{\Sigma}, \nu\}$.

$$l = \log(\prod_{j=1}^{n} p(\boldsymbol{x}_j, t_j, \boldsymbol{z}_j, \boldsymbol{æ}_j)) = \log(\prod_{j=1}^{n} p(\boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{æ}_j|t_j)p(t_j))$$

$$= \sum_{j=1}^{n}\{(\sum_{i=1}^{M}\sum_{l=1}^{k_i} z_{ji}æ_{jil}\log p(\boldsymbol{x}_j|g_i, c_{il})) + (\sum_{i=1}^{M}\sum_{l=1}^{k_i} z_{ji}æ_{jil}\log p(c_{il}|g_i)) + (\sum_{i=1}^{M} z_{ji}\log p(g_i))\}$$

$$= \sum_{j=1}^{n}\sum_{i=1}^{M}\sum_{l=1}^{k_i} z_{ji}æ_{jil}[\frac{1}{2}\log|\boldsymbol{\Sigma}|^{-1} - \frac{d}{2}\log 2\pi + \log\nu - \frac{\nu}{2}\log 2 - \log\Gamma(\frac{\nu}{2}) + (\frac{\nu}{2}-1)\log\nu +$$

$$(\frac{d}{2}+\frac{\nu}{2}-1)\log t_j - \frac{t_j}{2}(\nu+(\boldsymbol{x}_j-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}))] + \sum_{j=1}^{n}\sum_{i=1}^{M}\sum_{l=1}^{k_i} z_{ji}æ_{jil}\log p(c_{il}|g_i) + \sum_{j=1}^{n}\sum_{i=1}^{M} z_{ji}\log p(g_i)$$

Again, the EM algorithm alternates between averaging the loglikelihood, l, over the unobserved component membership, $\boldsymbol{æ}_j$, and scaling, t_j, and maximizing the averaged loglikelihood over the parameters and weights, $\boldsymbol{\Theta}$.

The component membership is regarded as a multinomial variable,

$$E(æ_{jil}|\mathcal{T}, \boldsymbol{\Theta}^{(s)}) = p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)} = \frac{p(c_{il}|g_i)^{(s)} f_{il}(\boldsymbol{x}_j|\boldsymbol{\mu}_{il}^{(s)}, \boldsymbol{\Sigma}^{(s)}, \nu^{(s)})}{\sum_{m=1}^{k_i} p(c_{im}|g_i)^{(s)} f_{im}(\boldsymbol{x}_j|\boldsymbol{\mu}_{im}^{(s)}, \boldsymbol{\Sigma}^{(s)}, \nu^{(s)})}.$$

Here $f_{il}(\boldsymbol{x}_j|\boldsymbol{\mu}_{il}^{(s)}, \boldsymbol{\Sigma}^{(s)}, \nu^{(s)})$ denotes the d-dimensional Student-t distribution with parameters $\{\boldsymbol{\mu}_{il}^{(s)}, \boldsymbol{\Sigma}^{(s)}, \nu^{(s)}\}$. Let us for simplicity denote the squared (scaled) Mahalanobis distance by $D_{jil}^{(s)}$, so that $D_{jil}^{(s)} = (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T(\boldsymbol{\Sigma}^{(s)})^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})$.

$$E(t_j|\mathcal{T}, \boldsymbol{\Theta}^{(s)}, \boldsymbol{æ}_j) = \frac{\nu^{(s)} + d}{\nu^{(s)} + (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T(\boldsymbol{\Sigma}^{(s)})^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})} = \frac{\nu^{(s)} + d}{\nu^{(s)} + D_{jil}^{(s)}}$$

$$E(\log t_j|\mathcal{T}, \boldsymbol{\Theta}^{(s)}) = DG(\frac{\nu^{(s)} + d}{2}) - \log 2 - \log\nu^{(s)} + D_{jil}^{(s)}$$

Here DG denotes the digamma function, $DG(x) = \frac{d\log\Gamma(x)}{dx}$. Inserting these relationships we find the expected complete data loglikelihood.

$$E(l|\mathcal{T}, \boldsymbol{\Theta}^{(s)}) = E(E(l|\mathcal{T}, \boldsymbol{\Theta}^{(s)}, (\boldsymbol{æ}_j)_{j=1}^{n}))$$

$$= n[\frac{1}{2}\log|\boldsymbol{\Sigma}|^{-1} - \frac{d}{2}\log 2\pi + \log\nu - \frac{\nu}{2}\log 2 - \log\Gamma(\frac{\nu}{2}) + (\frac{\nu}{2}-1)\log\nu + (\frac{d}{2}+\frac{\nu}{2}-1)(DG(\frac{\nu+d}{2}) - \log 2)]$$

$$-\sum_{j=1}^{n}\sum_{i=1}^{M}\sum_{l=1}^{k_i} z_{ji}p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)}\{(\frac{d}{2}+\frac{\nu}{2}-1)\log(\nu^{(s)} + D_{jil}^{(s)}) + \frac{1}{2}\frac{\nu^{(s)} + d}{\nu^{(s)} + D_{jil}^{(s)}}[\nu + (\boldsymbol{x}_j - \boldsymbol{\mu}_{il})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_{il})]\}$$

$$+\sum_{j=1}^{n}\sum_{i=1}^{M}\sum_{l=1}^{k_i} z_{ji}p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)}\log p(c_{il}|g_i) + \sum_{j=1}^{n}\sum_{i=1}^{M} z_{ji}\log p(g_i)$$

The maximization step (M-step) consists in maximizing the function $E(l|\mathcal{T}, \boldsymbol{\Theta}^{(s)})$ with respect to $\boldsymbol{\Theta}$ under the constraints that the mixing weights and the class weights sum to one, $\sum_{i=1}^{M} p(g_i) = 1$ and $\sum_{l=1}^{k_i} p(c_{il}|g_i) = 1$ for $i = 1, ..., M$.

At the sth step of the algorithm we define weights $w_{jil}^{(s)}$.

$$w_{jil}^{(s)} = p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)} \cdot \frac{\nu^{(s)} + d}{\nu^{(s)} + D_{jil}^{(s)}}$$

This leads to the following M-step equations.

$$\begin{aligned}
\boldsymbol{\mu}_{il}^{(s+1)} &= \frac{\sum_{j=1}^{n} z_{ji} w_{jil}^{(s)} \boldsymbol{x}_j}{\sum_{j=1}^{n} z_{ji} w_{jil}^{(s)}} \qquad for \;\; i = 1, ..., M \;\; and \;\; l = 1, ..., k_i \\
\boldsymbol{\Sigma}_{il}^{(s+1)} &= \frac{\sum_{j=1}^{n} z_{ji} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T}{\sum_{j=1}^{n} z_{ji} p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)}} \qquad for \;\; i = 1, ..., M \;\; and \;\; l = 1, ..., k_i \\
\boldsymbol{\Sigma}_{i}^{(s+1)} &= \frac{1}{n_i} \sum_{j=1}^{n} z_{ji} \sum_{l=1}^{k_i} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T \qquad for \;\; i = 1, ..., M \\
\boldsymbol{\Sigma}^{(s+1)} &= \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{M} z_{ji} \sum_{l=1}^{k_i} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_{il}^{(s)})^T \\
p(g_i) &= \frac{n_i}{n} \qquad for \;\; i = 1, ..., M \\
p(c_{il}|g_i)^{(s+1)} &= \frac{1}{n_i} \sum_{j=1}^{n} z_{ji} p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)} \qquad for \;\; i = 1, ..., M \;\; and \;\; l = 1, ..., k_i
\end{aligned}$$

The M-step update $\nu^{(s+1)}$ maximizing the expected total data loglikelihood with respect to ν.

$$\begin{aligned}
\nu^{(s+1)} = \underset{\nu}{\operatorname{argmax}} \{ &\nu \log \frac{\nu}{2} - 2 \log \Gamma(\frac{\nu}{2}) + \nu [DG(\frac{\nu^{(s)} + d}{2}) - \log 2] \\
&- \frac{\nu}{n} \sum_{j=1}^{n} \sum_{i=1}^{M} \sum_{l=1}^{k_i} z_{ji} p(c_{il}|g_i, \boldsymbol{x}_j)^{(s)} [\log(\nu^{(s)} + D_{jil}^{(s)}) + \frac{\nu^{(s)} + d}{\nu^{(s)} + D_{jil}^{(s)}}]\}
\end{aligned}$$

B.3 Class-Conditional Mixture with Common Component Distributions for the Classes

For common scale matrices for all components, we have the following.

$$\boldsymbol{X}_j | t_j, \boldsymbol{z}_j, \boldsymbol{\ae}_j \sim N_d(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}/t_j) \qquad t_j \sim \chi_{\nu}^2/\nu$$

$$t_j | \boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{\ae}_j \sim \chi_{\nu+d}^2 / [\nu + (\boldsymbol{x}_j - \boldsymbol{\mu}_l)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_l)]$$

The training set consists of observations and encoded class memberships, $\mathcal{T} = (\boldsymbol{x}_j, \boldsymbol{z}_j)_{j=1}^n$.

The derivation is essentially identical to that of the previous section. So, here only the M-step equations are presented.

For common scale matrices let us denote the squared (scaled) Mahalanobis distance from observation $\boldsymbol{x}_j$ to the location of component c_l by $D_{jl}^{(s)} = (\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})^T(\boldsymbol{\Sigma}^{(s)})^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})$. We then have the following equations for the probabilities and weights,

$$
\begin{aligned}
p(c_l|g_i, \boldsymbol{x}_j)^{(s)} &= \frac{p(c_l|g_i)^{(s)} f_l(\boldsymbol{x}_j|\boldsymbol{\mu}_l^{(s)}, \boldsymbol{\Sigma}^{(s)}, \nu^{(s)})}{\sum_{m=1}^{K} p(c_m|g_i)^{(s)} f_m(\boldsymbol{x}_j|\boldsymbol{\mu}_m^{(s)}, \boldsymbol{\Sigma}^{(s)}, \nu^{(s)})} \\
w_{jil}^{(s)} &= p(c_l|g_i, \boldsymbol{x}_j)^{(s)} \cdot \frac{\nu^{(s)} + d}{\nu^{(s)} + D_{jl}^{(s)}}
\end{aligned}
$$

For specific scale matrices for each component $\boldsymbol{\Sigma}^{(s)}$ is substituted by $\boldsymbol{\Sigma}_l^{(s)}$.

The M-step equations are the following.

$$
\begin{aligned}
\boldsymbol{\mu}_l^{(s+1)} &= \frac{\sum_{j=1}^{n}\sum_{i=1}^{M} z_{ji} w_{jil}^{(s)} \boldsymbol{x}_j}{\sum_{j=1}^{n}\sum_{i=1}^{M} z_{ji} w_{jil}^{(s)}} \qquad for \quad l = 1, ..., K \\
\boldsymbol{\Sigma}_l^{(s+1)} &= \frac{\sum_{j=1}^{n}\sum_{i=1}^{M} z_{ji} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})^T}{\sum_{j=1}^{n}\sum_{i=1}^{M} z_{ji} p(c_l|g_i, \boldsymbol{x}_j)^{(s)}} \qquad for \quad l = 1, ..., K \\
\boldsymbol{\Sigma}^{(s+1)} &= \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{M} z_{ji} \sum_{l=1}^{K} w_{jil}^{(s)} (\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})(\boldsymbol{x}_j - \boldsymbol{\mu}_l^{(s)})^T \\
p(g_i) &= \frac{n_i}{n} \qquad for \quad i = 1, ..., M \\
p(c_l|g_i)^{(s+1)} &= \frac{1}{n_i}\sum_{j=1}^{n} z_{ji} p(c_l|g_i, \boldsymbol{x}_j)^{(s)} \qquad for \quad i = 1, ..., M \quad and \quad l = 1, ..., K
\end{aligned}
$$

The M-step update $\nu^{(s+1)}$ to maximize the expected total data loglikelihood with respect to ν.

$$
\nu^{(s+1)} = \operatorname*{argmax}_{\nu}\{\nu \log\frac{\nu}{2} - 2\log\Gamma(\frac{\nu}{2}) + \nu[DG(\frac{\nu^{(s)} + d}{2}) - \log 2]
$$
$$
-\frac{\nu}{n}\sum_{j=1}^{n}\sum_{i=1}^{M}\sum_{l=1}^{K} z_{ji} p(c_l|g_i, \boldsymbol{x}_j)^{(s)}[\log(\nu^{(s)} + D_{jl}^{(s)}) + \frac{\nu^{(s)} + d}{\nu^{(s)} + D_{jl}^{(s)}}]\}
$$

Notice that by removing the class index i and summations thereof we have the EM algorithm for a one level multivariate Student-t mixture.

References

Andrews, D. F. and Herzberg, A. M. (1985), *Data, A Collection of Problems from Many Fields for the Student and Research Worker*, New York: Springer-Verlag.

Arslan, O., Constable, P. D. L., and Kent, J. T. (1993), “Domains of convergence for the EM algorithm: a cautionary tale in a location estimation problem”, *Statistics and Computing*, 3, 103–108.

Becker, R. A., Chambers, J. M., and Wilks, A. R. (1988), *The NEW S Language*, Pacific Grove, California: Wadsworth & Brooks/Cole.

Chambers, J. M. and Hastie, T. J., editors (1992), *Statistical Models in S*, Pacific Grove, California: Wadsworth & Brooks/Cole.

Cornish, E. A. (1954), "The multivariate t-distribution associated with a set of normal standard deviates", *Australian Journal of Physics*, 7, 531–542, cited in Lange et al. (1989).

Crawley, D. R. (1979), "Logistic discrimination as an alternative to Fisher's linear distriminant function", *N.Z. Statist.*, 14, 21–25, cited in Seber (1984) and McLachlan (1992).

Dasarathy, B. V. (1991), *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, Los Alamitos, California: IEEE Computer Society Press.

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

Duda, R. O. and Hart, P. E. (1973), *Pattern Classification and Scene Analysis*, New York: Wiley.

Dunnet, C. W. and Sobel, M. (1954), "A bivariate generalization of Student's t-distribution with tables for certain special cases", *Biometrika*, 41, 153–169, cited in Lange et al. (1989).

Efron, B. (1975), "The efficiency of logistic regression compared to normal discriminant analysis", *Journal of the American Statistical Assosiation*, 70, 892–898.

Fix, E. and Hodges, J. L. (1951), "Discriminatory analysis - nonparametric discrimination: consistency properties", Report Number 4, U.S. Air Force School of Aviation Medicine, Randolf Fields, Texas, reprinted in Dasarathy (1991).

Friedman, J. H. (1994), "Adaptive techniques for machine learning and function approximation", in Cherkassky, V., Friedman, J. H., and Wechsler, H., editors, *From Statistics to Neural Networks. Theory and Pattern Recognition Applications*, ASI Proceedings, Series F: Computer and Systems Sciences, pages 1–82, New York: Springer-Verlag.

Furman, W. D. and Lindsay, B. G. (1994), "Testing for the number of components in a mixture of normal distributions using moment estimators", *Computational Statistics & Data Analysis*, 17, 473–492.

Gath, I. and Geva, A. B. (1989), "Unsupervised optimal fuzzy clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 773–781.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986), *Robust Statistics, The Approach Based on Influence Functions*, New York: Wiley.

Hand, D. J. (1982), *Kernel Discriminant Analysis*, Chichester: Research Studies Press.

Hartigan, J. A. and Wong, M. A. (1979), "A K-means clustering algorithm", *Applied Statistics*, 28, 100–108.

Hastie, T., Buja, A., and Tibshirani, R. (1995), "Penalized discriminant analysis", *The Annals of Statistics*, in press.

Hastie, T. and Tibshirani, R. (1995), "Discriminant analysis by Gaussian mixtures", *Journal of the Royal Statistical Society, Series B*, in press, available via WWW at URL http://playfair.standford.edu/reports/hastie.

Hastie, T., Tibshirani, R., and Buja, A. (1994), "Flexible discriminant analysis by optimal scoring", *Journal of the American Statistical Association*, 89, 1255–1270.

Hjort, N. L. (1994), "The exact amount of t-ness that the normal model can tolerate", *Journal of the American Statistical Association*, 89, 665–675.

Huber, P. J. (1981), *Robust Statistics*, New York: Wiley.

Ingrassia, S. (1992), "A comparison between the simulated annealing and the EM algorithms in normal mixture decompositions", *Statistics and Computing*, 2, 203–211.

Johnsen, N. L. and Kotz, S. (1972), *Distributions in Statistics: Continuous Multivariate Distributions*, New York: Wiley.

Johnson, R. A. and Wichern, D. W. (1982), *Applied Multivariate Statistical Analysis*, Englewood Cliffs, New Jersey: Prentice-Hall, 2nd edition.

Kent, J. T., Tyler, D. E., and Vardi, Y. (1994), "A curious likelihood identity for the multivariate t-distribution", *Communications in Statistics, Simulation and Computation*, 23, 441–453.

Kohonen, T. (1990), "The self-organizing map", *Proceedings of the IEEE*, 78, 1464–1480, reprinted in Lau (1992).

Kohonen, T., Kangas, J., Laaksonen, J., and Torkkola, K. (1992), *LVQ PAK. The learning vector quantization package version 2.1*, LVQ Programming Team of the Helsinki University of Technology and Laboratory of Computer and Information Science, Espoo.

Langaas, M. (1995a), "Discrimination and classification", Technical Report Statistics 1/95, Department of Mathematical Sciences, The Norwegian Institute of Technology, The University of Trondheim.

Langaas, M. (1995b), "Mixture Codebook Classification, Part 1: Method outline", Technical Report Statistics 3/95, Department of Mathematical Sciences, The Norwegian Institute of Technology, The University of Trondheim.

Lange, K. L., Little, R. J. A., and Taylor, J. M. G. (1989), "Robust statistical modeling using the t distribution", *Journal of the American Statistical Association*, 84, 881–896.

Lippmann, R. P. (1994), "Neural networks, Bayesian a posteriori probabilities, and pattern classification", in Cherkassky, V., Friedman, J. H., and Wechsler, H., editors, *From Statistics to Neural Networks. Theory and Pattern Recognition Applications*, NATO ASI Series F: Computer and Systems Sciences, vol. 136, pages 83–104, New York: Springer-Verlag.

Little, R. A. J. and Rubin, D. B. (1987), *Statistical Analysis with Missing Data*, New York: Wiley.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis*, London: Academic Press.

Maronna, R. A. (1976), "Robust M-estimators of multivariate location and scatter", *The Annals of Statistics*, 4, 51–67.

McLachlan, G. J. (1987), "On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture", *Applied Statistics*, 36, 318–324.

McLachlan, G. J. (1992), *Discriminant Analysis and Statistical Pattern Recognition*, New York: John Wiley & Sons.

McLachlan, G. J. and Basford, K. E. (1988), *Mixture Models, Inference and Applications to Clustering*, New York: Marcel Dekker.

Moody, J. E. and Darken, C. J. (1989), "Fast learning in networks of locally-tuned processing units", *Neural Computation*, 1, 281–294.

Panel on Discriminant Analysis, Classification and Clustering (1989), "Discriminant analysis and clustering", *Statistical Science*, 4, 34–69.

Peck, R., Fisher, L., and Van Ness, J. (1989), "Approximate confidence intervals for the number of clusters", *Journal of the American Statistical Association*, 84, 184–191.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988), *Numerical Recipes in C*, Cambridge: Cambridge University Press, 1st edition.

Redner, R. A. and Walker, H. F. (1984), "Mixture densities, maximum likelihood and the EM algorithm", *SIAM Review*, 26, 195–239.

Ripley, B. D. (1987), *Stochastic Simulation*, New York: John Wiley & Sons.

Ripley, B. D. (1994), "Neural networks and related methods for classification (with discussion)", *Journal of the Royal Statistical Society, Series B*, 56, 409–456.

Roeder, K. (1994), "A graphical technique for determining the number of components in a mixture of normals", *Journal of the American Statistical Association*, 89, 487–495.

Rubin, D. B. (1983), "Iteratively reweighted least squares", in Kotz, S. and Johnson, N. L., editors, *Encyclopedia of Statistical Sciences 4*, pages 272–275, New York: Wiley.

Seber, G. A. F. (1984), *Multivariate Observations*, New York: Wiley.

Specht, D. F. (1990), "Probabilistic neural networks", *Neural Networks*, 3, 109–118.

Taxt, T., Hjort, N. L., and Eikvil, L. (1991), "Statistical classification using a linear mixture of two multinormal probability densities", Technical Report 5/1991, Institute of Mathematics, University of Oslo.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985), *Statistical Analysis of Finite Mixture Distributions*, Chichester: John Wiley & Sons.

Tråvén, H. G. C. (1991), "A neural network approach to statistical pattern classification by "semiparametric" estimation of probability density functions", *IEEE Transactions on Neural Networks*, 2, 366–377.

Venables, W. N. and Ripley, B. D. (1994), *Modern Applied Statistics with S-Plus*, New York: Springer-Verlag.

Wabha, G. (1994), "Generalization and regularization in nonlinear learning systems", Technical Report 921, Department of Statistics, University of Wisconsin, Madison.

Wong, M. A. (1985), "A bootstrap testing procedure for investigating the number of subpopulations", *Journal of Statistical Computation and Simulation*, 22, 99–112.

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

Xu, L., Krzyzak, A., and Yuille, A. (1994), "On radial basis function nets and kernel regression: statistical consistency, convergence rates, and receptive field size", *Neural Networks*, 7, 609–628.