Discrimination and Classification

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

Abstract

The aim of this report is to present methods from statistics, neural networks, non-parametric regression and pattern recognition to perform discrimination and classification. The methods are compared on theoretical and empirical grounds to highlight strengths and weaknesses. A common platform for classification is also outlined. The emphasis is on multiple (more than two) classes.

Some keywords: Supervised Classification; Discriminant Analysis; Multiple Classes; Multi-layer Perceptrons (MLP).

Contents

1 Introduction

The aim of this report is to present and empirically compare methods for discrimination and classification with multiple classes. The methods considered are statistical regression, discrimination and classification methods along with the connectionists multilayer perceptron. The comparison is based on two data sets; one data set from biology and one synthetic data set. A common platform for multiple group classification is also outlined.

A variety of new statistical methods for regression and classification have been developed on the basis of nonparametric regression over the last few years. These methods include projection pursuit regression of Friedman and Stuetzle (1981), multivariate regression splines (MARS) of Friedman (1991) and generalized additive methods (GAM) of Hastie and Tibshirani (1990). Multilayer perceptrons (MLP) are a popular tool for classification and have in general given good results on various problems. A broad introduction to neural networks is Hertz et al. (1991) and practical issues are given thorough treatment in Hush and Horne (1993). For the view of a statistician see Ripley (1993a, 1993b and 1994b) and Cheng and Titterington (1994).

The empirical results of this report are based on the two above mentioned data sets. The first data set consists of in vivo absorption spectra from marine phytoplankton from a survey of 31 low- and high-light adapted phytoplankton species presented in Johnsen et al. (1994). The second data set is a synthetic data set consisting of three classes, where each class is an equal mixture of two Gaussian distributions.

The statistical analyses in this report 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). Part of the MLP analyses are done using the software Aspirin/MIGRAINES 6.0 from the Mitre Corporation, Leighton (1992).

2 Classification

2.1 Decision Theoretic Framework

This presentation of the decision theoretic framework is based upon Wald's classical decision theory, as presented for example in Casella and Berger (1990). See also chapter 2 of Duda and Hart (1973) and Hjort (1986).

An observation is a feature vector $\boldsymbol{X} = \boldsymbol{x}$ from some space $\mathcal{X}$, $\boldsymbol{x} \in \mathcal{X}$, the *sample space*. The true class membership is $\omega \in \Omega = \{\omega_1, \omega_2, ..., \omega_M\}$, M being the total number of classes (M is finite). This is the true state of nature. We assume that pairs $(\boldsymbol{X}, \omega)$ are drawn independently from the joint probability distribution $p(\boldsymbol{x}, \omega)$ on $\mathcal{X} \times \Omega$.

Given an observation (vector of features), $\boldsymbol{X} = \boldsymbol{x}$, we want to make a decision regarding the true class of the observation. This is our *classification problem*; predict the true class ω for an observation $\boldsymbol{X} = \boldsymbol{x}$. The *action space* $\mathcal{Y}$ is the space of allowable decisions to our classification problem. An element of $\mathcal{Y}$ is called y. The action space can in general be seen as an extension of the space of the classes, $\mathcal{Y} = \{\Omega, \text{'}outlier\text{'}, \text{'}doubt\text{'}, ...\}$, as in Ripley (1994a). In this presentation we assume, however, that the action space equals the space of the classes, $\mathcal{Y} = \Omega$.

The relationship between the action space $\mathcal{Y}$ and the space of the classes Ω is quantified by the *loss function* $L(\omega, y)$, which gives the loss incurred if ω is the true state of nature and action y is taken. A loss function is a function $L\colon \Omega \times \mathcal{Y} \rightarrow \mathbb{R}$.

The link between the sample space and the action space is the *classifier*. The classifier is a nonrandomized decision rule δ from the space of allowable decisions that specifies, for each observation $\boldsymbol{x} \in \mathcal{X}$, what action $y \in \mathcal{Y}$ to take if $\boldsymbol{X} = \boldsymbol{x}$ is observed. The classifier is a map from the sample space $\mathcal{X}$ to the action space $\mathcal{Y}$, $\delta : \mathcal{X} \rightarrow \mathcal{Y}$.

A classifier δ can be evaluated by its *risk function* $R(\omega, \delta)$, given by,

$$R(\omega, \delta) = E_{\boldsymbol{x}|\omega} L(\omega, \delta(\boldsymbol{x})) = \int_{\mathcal{X}} L(\omega, \delta(\boldsymbol{x})) p(\boldsymbol{x} \mid \omega) d\boldsymbol{x}. \tag{1}$$

The risk function is the expected loss by using the classifier δ when ω is the true state of nature. It is desirable to choose a classifier that has a small value of the risk function $R(\omega, \delta)$ for all classes ω. Comparing classifiers on the basis of their respective risk functions can be difficult since different classifiers might give superior results to others in separate subspaces of Ω. In Bayes decision analysis one attempts to choose the classifier δ that minimizes the risk on the space of the classes, Ω, weighed by the prior probabilities of the classes. This is called the Bayes risk, $B(\pi, \delta)$, and is given as,

$$B(\pi, \delta) = \sum_{\omega \in \Omega} R(\omega, \delta) \pi(\omega) \tag{2}$$

where $\pi(\omega)$ is the prior probability given to class ω.

Minimizing the Bayes risk is achieved by for each $\boldsymbol{x}$ choosing the classifier $\delta(\boldsymbol{x})$ that minimizes the posterior expected loss $E_{\omega|\boldsymbol{x}}(L(\omega, \delta(\boldsymbol{x}))$,

$$E_{\omega|\boldsymbol{x}}(L(\omega, \delta(\boldsymbol{x})) = \sum_{\omega \in \Omega} L(\omega, \delta(\boldsymbol{x})) p(\omega \mid \boldsymbol{x}) \tag{3}$$

where $p(\omega \mid \boldsymbol{x})$ is the posterior probability for class ω given $\boldsymbol{x}$. In Duda and Hart (1973) $E_{\omega|\boldsymbol{x}}(L(\omega, \delta(\boldsymbol{x}))$ is called the *conditional risk* and is the defining equation for what is called the Bayes decision rule and the minimum Bayes risk is simply called the Bayes risk.

Using the zero-one loss function,

$$L_{0/1}(\omega, y) = \begin{cases} 0 & \text{if } \omega = y \\ 1 & \text{otherwise} \end{cases}$$

the classifier is constructed by choosing the class giving the maximum posterior probability. We will call this *MAP classification* (maximum a posteriori).

In classification the aim is to predict the true class ω for an observation $\boldsymbol{X} = \boldsymbol{x}$. In discrimination the aim is to separate the sample space into disjoint regions for the classes $\{\omega_1, ..., \omega_M\}$. These two tasks are closely related. Both classification and discrimination put strong emphasis on the posterior distribution of the class, ω, conditional on the observation, $\boldsymbol{x}$, $p(\omega|\boldsymbol{x})$.

2.2 Allocation Principles

In the decision theoretic framework the allocation of new observations (classification) is based on minimized Bayes risk. Below is a list of some popular allocation principles, see Seber (1984) chapter 6.

Minimize the (total) expected cost of misclassification (TCM) A loss function is used to assign a cost to each possible misclassification. This is equivalent to minimizing the Bayes risk with a loss function with zero loss for correct classification and specific loss for each misclassification. The rule is to classify an observation $\boldsymbol{x}$ to class ω_k if

$$\sum_{\substack{l=1\\l\neq i}}^{M} L(\omega_l,\omega_i)\pi(\omega_l)p(\boldsymbol{x}|\omega_l) \geq \sum_{\substack{l=1\\l\neq k}}^{M} L(\omega_l,\omega_k)\pi(\omega_l)p(\boldsymbol{x}|\omega_l) \qquad for \quad \forall i \neq k.$$

Minimize the total probability of misclassification The rule is to classify an observation $\boldsymbol{x}$ to class ω_k if

$$\pi(\omega_i)p(\boldsymbol{x}|\omega_i) \leq \pi(\omega_k)p(\boldsymbol{x}|\omega_k) \qquad for \quad \forall i \neq k$$

This is equivalent to minimizing the Bayes risk with zero-one loss function.

Minmax allocation Minimize the maximum (individual) probability of misclassification for the classes. The method is used for giving more attention to rare classes, see Seber (1984) section 6.2.

Likelihood ratio When the prior probabilities are unknown an intuitive rule is to classify to the class that maximizes the likelihood function for the new observation. This corresponds to using uniform priors and minimizing the Bayes risk with zero-one loss function.

Minimize the within-group variance to the between group variance In 1936 R.A. Fisher developed the classical method of linear discriminant analysis. His aim was to find a linear combination of the explanatory variables which maximized the ratio of its between-group to within-group variance, see for example Johnson and Wichern (1982), chapter 11. This is discussed further in section 4.1.

MAP classification is by far the most popular classification rule. A more realistic classifier is based on minimizing the TCM with different costs for the misclassifications, that is, to rank the mistakes in order of seriousness. (For example giving higher cost to classify a serious disease as a mild one than vice versa.) This also opens for assigning zero loss to certain misclassifications.

2.3 Discriminant Functions

It is often convenient to base the construction of a classifier on a set of *discriminant functions*, $g_i(\boldsymbol{x})$, $i = 1, ..., M$.

If the action space $\mathcal{Y}$ equals the space of the classes, Ω, the classifier may assign a feature $\boldsymbol{x}$ to the class ω_i if

$$g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \qquad \forall j \neq i \tag{4}$$

This is what is done using Fisher's linear discriminant functions in section 4.1.

Formally, let the classifier $\delta : \mathcal{X} \to \mathcal{Y}$, be divided into two separate steps, $\boldsymbol{g}$ and h to give the composition $\delta(\boldsymbol{x}) = h(\boldsymbol{g}(\boldsymbol{x}))$. The first step is the construction of a set of discriminant functions $g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), ..., g_M(\boldsymbol{x})$ joined in a discriminant vector $\boldsymbol{g}(\boldsymbol{x})$. The function $\boldsymbol{g}$ is a mapping from the sample space $\mathcal{X}$ to the *space of discriminants* $\mathcal{D}$, $\boldsymbol{g} : \mathcal{X} \to \mathcal{D}$. This can be regarded as a transformation from the sample space to the discriminant space. The second step is the mapping h which takes the discriminants into the action space, $h : \mathcal{D} \to \mathcal{Y}$. This map does not need to be one-to-one.

$$
\begin{tikzcd}
\mathcal{X} \arrow[rr, "\delta"] \arrow[dr, "\boldsymbol{g}"'] & & \mathcal{Y} \\
& \mathcal{D} \arrow[ur, "h"'] &
\end{tikzcd}
$$

The classes, $\omega \in \Omega$, can be encoded as targets $\boldsymbol{t}$ from the *target space* $\mathcal{T}$ by a one-to-one encoding W. The encoding can be any one-to-one mapping from Ω to $\mathcal{T}$, $W : \Omega \to \mathcal{T}$. One commonly used encoding is the 0/1 indicator variable encoding of the classes. Let t_i be a (dichotomous) indicator variable denoting class ω_i or not.

$$t_i = \begin{cases} 1 & \text{if } \omega = \omega_i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Then the target space $\mathcal{T}$ equals $\{0, 1\}^M$. Another possibility is to encode

$$t_i = \begin{cases} a & \text{if } \omega = \omega_i \\ b & \text{otherwise} \end{cases}$$

yielding $\mathcal{T} = \{a, b\}^M$. In the remainder of this section we assume that targets are given in equation (5).

In the neural network community a popular choice for the h function is to let h be to choose the class encoded by the target $\boldsymbol{t}$ with minimum Euclidean distance to the discriminant function $\boldsymbol{g}$.

$$\begin{aligned} \min_{\boldsymbol{t}} \parallel \boldsymbol{t} - \boldsymbol{g} \parallel^2 &= \min_{\boldsymbol{t}} \sum_{j=1}^{M} (t_j - g_j)^2 = \min_{\boldsymbol{t}} \sum_{j=1}^{M} (t_j^2 - 2t_j g_j + g_j^2) \\ &= 1 - 2g_m + \sum_{j=1}^{M} g_j^2 = 1 - 2g_m + \parallel \boldsymbol{g} \parallel^2 \end{aligned} \tag{6}$$

Here g_m denotes the maximal element in $\boldsymbol{g}$. We see that classifying to the class corresponding to the indicator variable target $\boldsymbol{t}$ with minimum Euclidean distance to the discriminant function $\boldsymbol{g}$ is the same as classifying to the class represented by the discriminant function with the largest value in $\boldsymbol{g}$, Ripley (1993a).

Instead of working with the loss function $L : \Omega \to \mathcal{Y}$ it would be convenient to define a new loss function L_h^* in terms of the targets $\boldsymbol{t}$, the discriminant functions $\boldsymbol{g}$ and the mapping h.

$$L(\omega, y) = L(\omega, \delta(\boldsymbol{x})) = L(W^{-1}(\boldsymbol{t}), h(\boldsymbol{g}(\boldsymbol{x})) = L_h^*(\boldsymbol{t}, \boldsymbol{g}(\boldsymbol{x}))$$

The subscript h denotes that the mapping h is used from the discriminant space to the action space. This will open for the construction of loss functions $L_h^*(\boldsymbol{t}, \boldsymbol{g})$, $L_h^* : \mathcal{T} \times \mathcal{D} \to \mathbb{R}$, from a wider class than what is natural for the loss functions, $L(\omega, \delta)$, $L : \Omega \times \mathcal{Y} \to \mathbb{R}$. The zero-one loss function $L_{0/1}$ will correspond to the zero-one loss function $L_{h0/1}^*$:

$$L_{0/1}(\omega, \delta(\boldsymbol{x})) = \begin{cases} 0 & \text{if } \omega = \delta(\boldsymbol{x}) \\ 1 & \text{otherwise} \end{cases} \Leftrightarrow L_{h0/1}^*(\boldsymbol{t}, \boldsymbol{g}(\boldsymbol{x})) = \begin{cases} 0 & \text{if } W^{-1}(\boldsymbol{t}) = h(\boldsymbol{g}(\boldsymbol{x})) \\ 1 & \text{otherwise} \end{cases}$$

The Euclidean loss function L_h^* is not possible in general to translate to the loss function L because of the fact that the mapping h need not be one-to-one and therefore h^{-1} is not unique.

$$L_h^*(\boldsymbol{t}, \boldsymbol{g}(\boldsymbol{x})) = \| \boldsymbol{t} - \boldsymbol{g}(\boldsymbol{x}) \|^2 \Rightarrow L(\omega, \delta(\boldsymbol{x})) = \| W(\omega) - h^{-1}(\delta(x)) \|^2$$

Since the function h need not be one-to-one this is not well defined. But still, working with the loss function L_h^* is preferable in settings such as the multilayer perceptron in section 3 and using regression to construct discriminant functions in section 4.5.

2.4 Constructing Classifiers

We have seen that one possibility for classification is to choose a classifier that is optimal with respect to minimizing the Bayes risk with a chosen loss function. The construction of this classifier requires that the posterior probability distribution $p(\omega|\boldsymbol{x})$ or the prior probabilities $\pi(\omega)$ and the class-conditional distributions $p(\boldsymbol{x} \mid \omega)$ are known. This is seldom the case. Therefore only classifiers that are *approximations* to the Bayes decision rule can be found.

The *training set* is a data set of observation of feature vectors together with the true class membership for the observations. Let there be a total of n observations from a total of M classes. The training set is used to construct the classifier. Let $\tau = \{(\boldsymbol{x}^1, \boldsymbol{t}^1), ..., (\boldsymbol{x}^n, \boldsymbol{t}^n)\}$ denote the training set. The number of observation for class ω_i is denoted n_i (equivalently, n_t denotes the number of observation for target $\boldsymbol{t}$).

The most popular approach to classification is to use the Bayes decision rule with the zero-one loss function. Two approaches for estimating the Bayes decision rule classifier are listed below, based on the terminology of Dawid (1976),

1. The *diagnostic paradigm* approach. We here focus on parameterizing the posterior distribution for the classes, $p(\omega|\boldsymbol{x})$. We estimate the posterior probability distribution $\hat{p}(\omega \mid \boldsymbol{x})$ and classify to the class ω giving $\max_{i=1,...,M} \hat{p}(\omega_i \mid \boldsymbol{x})$.

2. The *sampling paradigm* approach. There focus is on the prior probabilities for the classes and on the class-conditional distributions, $p(\boldsymbol{x}|\omega)$. Estimate the prior probabilities for the classes, $\hat{\pi}(\omega_i)$, and the class-conditional distribution of the observations, $\hat{p}(\boldsymbol{x}|\omega_i)$, for $i = 1, ..., M$. Then classify to the class ω with $\max_{i=1,...,M} \hat{\pi}(\omega_i)\hat{p}(\boldsymbol{x} \mid \omega_i)$.

For loss functions, L, different from the zero-one loss function the diagnostic approach can be taken; estimate the posterior distribution and choose the appropriate feature.

Another method for constructing a classifier that estimates the Bayes decision rule is to base the construction on the discriminant functions of the previous section. Then the classifier $\delta(\boldsymbol{x}) = h(\boldsymbol{g}(\boldsymbol{x}))$ can be constructed by minimizing an estimate of the Bayes risk directly.

Let $\boldsymbol{g}$ be the output from the discriminant functions, $\boldsymbol{g} = \boldsymbol{g}(\boldsymbol{x})$. An estimator for the Bayes risk (2) based on the discriminant functions $\boldsymbol{g}$ is the error measure E_{class}:

$$E_{class} = \sum_{\boldsymbol{t}} \hat{\pi}(\boldsymbol{t}) \sum_{l:\boldsymbol{t}^l=\boldsymbol{t}} \hat{p}(\boldsymbol{x}^l \mid \boldsymbol{t}^l) L_h^*(\boldsymbol{t}^l, \boldsymbol{g}(\boldsymbol{x}^l)) = \sum_{\boldsymbol{t}} \frac{n_t}{n} \sum_{l:\boldsymbol{t}^l=\boldsymbol{t}} \frac{1}{n_t} L_h^*(\boldsymbol{t}^l, \boldsymbol{g}(\boldsymbol{x}^l))$$

$$= \sum_{\boldsymbol{t}} \frac{n_t}{n} \cdot \frac{1}{n_t} \sum_{l:\boldsymbol{t}^l=\boldsymbol{t}} L_h^*(\boldsymbol{t}^l, \boldsymbol{g}(\boldsymbol{x}^l)) = \frac{1}{n} \sum_{\boldsymbol{t}} \sum_{l:\boldsymbol{t}^l=\boldsymbol{t}} L_h^*(\boldsymbol{t}^l, \boldsymbol{g}(\boldsymbol{x}^l)) = \frac{1}{n} \sum_{l=1}^{n} L_h^*(\boldsymbol{t}^l, \boldsymbol{g}(\boldsymbol{x}^l)) \tag{7}$$

Here the class priors are estimated by $\hat{\pi}(\boldsymbol{t}) = n_t/n$ and $\hat{p}(\boldsymbol{x} \mid \boldsymbol{t})$ is the empirical distribution of the observed feature vectors $\boldsymbol{x}$ with target $\boldsymbol{t}$, $\hat{p}(\boldsymbol{x} \mid \boldsymbol{t}) = 1/n_t$. Assuming equal class priors $\hat{\pi}(\boldsymbol{t}) = 1/M$ leads to the modified error measure E^*_{class}:

$$E^*_{class} = \sum_{\boldsymbol{t}} \frac{1}{M} \sum_{l:\boldsymbol{t}^l \in \boldsymbol{t}} \frac{1}{n_t} L_h^*(\boldsymbol{t}^l, \boldsymbol{g}) \tag{8}$$

With quadratic loss function the error measure E_{class} is called the average squared residual, ASR:

$$ASR = \frac{1}{n} \sum_{l=1}^{n} \| \boldsymbol{t}^l - \boldsymbol{g}^l \|^2 \tag{9}$$

The ASR is sometimes called the residual mean squared error RMSE (not to be confused with the root mean squared error also denoted RMSE).

This presentation is based on that a specified form is assumed for the discriminant functions $\boldsymbol{g}$ (parametric, semi-parametric or nonparametric). The form of $\boldsymbol{g}$, h and how to choose the loss function L_h^*, is the issue of many of the methods for discrimination and classification presented in the following sections.

2.5 Evaluation Principles

Classifiers have to be estimated and evaluated using available observations of features and correct class memberships. It is well known that for instance the error rate is underestimated when using the same observations to construct the classifier and to calculate the error rate, see Efron (1986). The total data set of classified observation can be divided into three data sets; *training set, test set* and *validation set.* The training set is used to construct the classifier (as seen in the previous section). The validation set is used to find the optimal model complexity and model-parameters for the classifier (for instance the network size or the number of terms in a regression). These parameters can be called *meta-parameters.* The test set is used to evaluate the performance and generalization ability of the classifier and to form a basis for comparison between different classifiers. With scarce total data these data sets might be small. Then the data set is often divided in two only; one training set and one test set. The test set is then used to perform the duties of both the test and the validation set, but will bias the comparisons.

Another solution to the problem with scarce data is to use v-fold or leave-one-out cross-validation, see the review of Stone (1978) and section 11.5 of Breiman et al. (1984). In v-fold cross-validation the training set is split into v parts. The classification rule is constructed using

$v-1$ of the parts and used to classify the part left out. The process is repeated v times and the error rates (misclassification rates) are averaged. Leave-one-out cross-validation equals v-fold cross-validation with v equal to the size of the training set. Choosing the meta parameters can be done using a *validation layer* of v-fold cross-validation, and compare the performance of the classifiers by using a second layer, *test layer*, of cross-validation.

In this presentation v-fold cross-validation is used for method comparisons on the data set of marine phytoplankton in section 5 and separate training, test and validation sets are used for the three class synthetic data set in section 6.

3 Multilayer Perceptrons

Neural networks can be divided into two categories; dynamic (among others recurrent networks, Hopfield nets) and static (among others multilayer perceptron, radial basis function networks, learning vector quantizer) neural networks. In addition the learning procedures are divided into supervised (with a teacher) and unsupervised (clustering) learning. A broad introduction to neural networks is Hertz et al. (1991) and Hush and Horne (1993) give a practical review of recent developments in supervised neural networks. Presentations from a statistical perspective are found in Ripley (1993a, 1993b and 1994b) and Cheng and Titterington (1994).

The most popular static network is the multilayer perceptron (MLP). A *perceptron*, Rosenblatt (1958) and Minsky and Papert (1988), is originally a function which computes a weighted sum of input variables and returns "1" if the weighted sum is positive and "-1" if it is negative, a so called hard-limiter. The multilayer perceptron is constructed from individual perceptrons which are called *nodes*. The nodes differ from the originally perceptrons in that a general *activation* or *transfer* function is use in place of the hard-limiter. The nodes are organized in a layered structure. There is one *input layer*, one or more *hidden layers* and one *output layer*. The number of hidden layers can be zero, then the network is called a single perceptron. The input layer consists of the explanatory (input) variables which are called input nodes. The hidden layers consist of hidden nodes and the output layer consists of output nodes. A threshold node, called a bias, is often added (or subtracted) to the weighted sum to the hidden and output nodes. There are one-way connections between the network layers. That is, information is fed to the input nodes and propagated forward in the network to the output nodes.

A network with one hidden layer and D input, H hidden and M output nodes is called a $D-H-M$ network. Figure 1 shows a MLP with one hidden layer.

Dealing with classification the popular approach is to represent each class with one output node. The exception is when there are two classes, then only one output node is used. The emphasis here is on more than two classes. The target $\boldsymbol{t} \in \mathcal{T}$ is usually the indicator variable encoding of the classes $\omega \in \Omega$ found in equation (5) (also called one-of-c coding), but often also the more general $\mathcal{T} = \{a, b\}^M$ encoding is used. The encoding is one-to-one. The output from the MLP is denoted $\boldsymbol{g}$. For a MLP with one hidden layer the ith component of $\boldsymbol{g}$ is

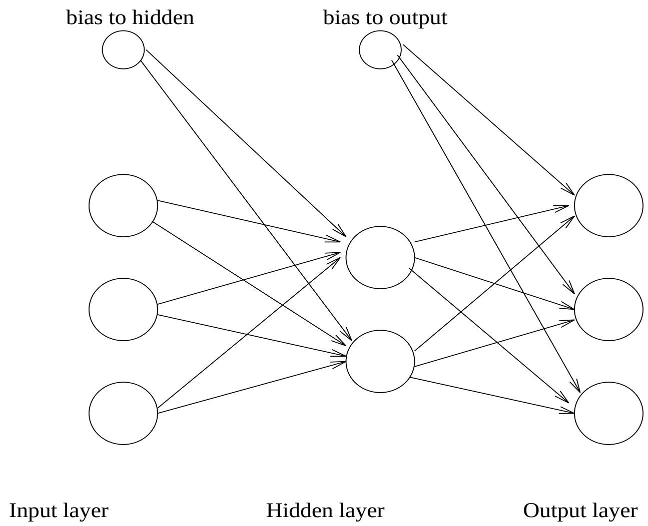

Figure 1: MLP with one hidden layer. The network has three input nodes, two hidden nodes and three output nodes and is called a 3-2-3 network.

given by the mathematical relationship for the ith output node,

$$g_i(\boldsymbol{x}) = f_o(\beta_{i0} + \sum_{l=1}^{H} \beta_{il} \cdot f_h(\alpha_{l0} + \sum_{s=1}^{D} \alpha_{ls} x_s)) \qquad i = 1, ..., M. \tag{10}$$

x_s	the feature presented to the sth input node (the sth element of the feature vector $\boldsymbol{x}$)
α_{ls}	weight from the sth input node to the lth hidden node α_{l0} is the bias to the lth hidden node
β_{il}	weight from the lth hidden node to the ith output node β_{i0} is the bias to the ith output node
f_h	activation function for the hidden layer
f_o	activation function for the output layer
D	the number of inputs (dimension of the sample space)
H	the number of hidden nodes
M	the number of output nodes

Different activation (transfer) functions are possible, the most popular ones are the linear activation function $f(u) = u$ and the sigmoid (logistic) activation function:

$$f(u) = \frac{1}{1+e^{-u}} - \frac{1}{2} \qquad or \qquad f(u) = \frac{1}{1+e^{-u}} = \frac{e^u}{e^u+1}$$

The given sigmoids are for output ranges $[-\frac{1}{2}, \frac{1}{2}]$ and [0,1] respectively. It is also possible to include skip layer connections from the input layer to the output layer. This is achieved by adding an extra term $\sum_{s=1}^{D} \gamma_{si} x_s$ to equation (10) inside the output activation function, where γ_{si} denotes the weight from the sth input node to the ith output node, see Ripley (1994b).

We can regard the output vector from the MLP as our discriminant vector. The components of the output vector are not necessarily restricted to lie in $[0, 1]$. To introduce a notion of probability a modified vector may be preferred. This ensures that the components in the output vector will lie in $[0, 1]$ and that they sum to one,

$$g_i := \frac{\exp g_i}{\sum_{j=1}^{M} \exp g_j}. \tag{11}$$

The MLP bases the construction of the classifier $\delta(\boldsymbol{x})$ on $\boldsymbol{g}(\boldsymbol{x})$, that is, $\delta(\boldsymbol{x}) = h(\boldsymbol{g}(\boldsymbol{x}))$ (as in section 2.1). The most common choice of h is to classify to the class represented by the discriminant function (output node) with the largest values in $\boldsymbol{g}$, see equation (6).

The weights of the MLP are estimated on the basis of the training set. The MLP can be *trained* (the weights can be estimated) using different variations of the *back-propagation algorithm.* The back-propagation algorithm is in its simplest form a gradient descent algorithm. The back-propagation algorithm was invented independently several times in the neural network community. See chapter 5 and 6 of Hertz et al. (1991) for early references to the back-propagation algorithm and for other training algorithms.

A loss function, L^*, is used to measure the discrepancy between the output $\boldsymbol{g}$ from the network and the target $\boldsymbol{t}$. The most popular loss function is the squared Euclidean distance (quadratic loss),

$$L^*(\boldsymbol{t}, \boldsymbol{g}) = \parallel \boldsymbol{t} - \boldsymbol{g} \parallel^2 .$$

Another possibility is to use Kullback-Leibler distance, Solla et al. (1988) and Bridle (1990). Then the targets and the outputs are viewed as probabilities (therefore the scaled outputs (11) are preferred).

$$L^*(\boldsymbol{t}, \boldsymbol{g}) = \sum_{i=1}^{M} (t_i \log \frac{t_i}{g_i} + (1 - t_i) \log \frac{1 - t_i}{1 - g_i})$$

The Kullback-Leibler distance is also called the entropy loss, and the use of the Kullback-Leibler distance for scaled outputs is called softmax, Bridle (1989 and 1990). The *error measure* is the sum of the loss function over all the observations in the data set and is denoted E_{MLP}.

$$E_{MLP} = \frac{1}{n} \sum_{l=1}^{n} L^*(\boldsymbol{t}^l, \boldsymbol{g}^l)$$

Comparing this with the approach in section 2 we see that the error measure is an estimate of the Bayes risk. That is, the error measure estimates the class priors by the class frequencies in the data set, see equation (7). Sometimes the class frequencies do not reflect the class priors and it is important to be aware of the assumptions made by using this error measure. If equal class priors are desired each observation should be weighted with the inverse of the class size, see equation (8).

The backpropagation algorithm gives a prescription for changing the weights in a MLP network in order to *learn* a training set of input and output pairs. The weights of the network are selected so that the chosen error measure is minimized. The error measure is differentiated with respect to all the weights in the network. This is done directly for the weights from the hidden to the output layer and indirectly using the chain rule for all other weights. The quadratic loss function is a differentiable function if the activation functions are differentiable. The network weights are changed by an amount proportional to the error signal,

$$\begin{aligned} w_{il}^{new} &= w_{il}^{old} + \Delta w_{il} \\ \Delta w_{il} &= -\eta \frac{\partial E_{MLP}}{\partial w_{il}} \end{aligned} \tag{12}$$

where w_{il} denotes a general weight in the network. The parameter η is called the *learning rate*. The above equations for updating the weights are written as sums over all the observations in the training set. Updating the weights after each presentation of the total training set is called *batch* update. One presentation of the training set is called an *epoch*. It is also possible to use the update rules incrementally, that is, to update the weights for each observation, *on-line* update. This is often done, Hertz et al. (1991). A solution in between is to update after each mth pattern (observation), $1 \leq m \leq n$. For non-batch update rules the observations can be presented in random or sequential order.

A momentum term can be used to improve the convergence of the back-propagation algorithm. The update equation (12) can cause slow convergence when η is small and can cause wide oscillations when η is too large, Hertz et al. (1991). A modified update equation may be preferred.

$$\Delta w_{il}(c+1) = -\eta \frac{\partial E_{MLP}}{\partial w_{il}} + \zeta \Delta w_{il}(c)$$

where $\Delta w_{il}(c+1)$ is the weight change at update step $c+1$, $\Delta w_{il}(c)$ is the weight change at update step c and ζ is the momentum parameter. Variations of the back-propagation algorithm include line search for optimal step size, conjugate gradient search directions and individual step size and momentum parameters for each weight (delta-bar-delta). See Schiffmann et al. (1994) for details and empirical comparisons between the different variations of back-propagation. Another possibility is to include second derivatives (Hessian matrix) in the calculations, see Ripley (1993a).

There are two main tasks connected to the construction of a MLP; finding the optimal network architecture and learning the optimal weights. There are growing and pruning algorithms that combine these two tasks. But the two tasks can be treated separately. In general, parameters associated with learning will influence the speed of convergence of the learning process and they can also have impact on the final network weights because one function can often be represented by more than one set of weights (not a unique representation). Important choices are how often to update the weights and when to stop training to avoid overfitting.

For model validation and evaluation the error measure can be a good estimate of the Bayes risk when used on validation or test sets (on separate test and validation sets or in v-fold cross-validation or in leave-one-out cross-validation). Criteria for network validation and evaluation are discussed for example in Barron and Barron (1988).

4 Statistical Methods for Discrimination and Classification

In this section we present a selection of statistical methods for discrimination and classification.

4.1 Linear and Quadratic Discriminant Analysis

Using the sampling paradigm, see section 2.4, the Bayes decision rule with zero-one loss function is to choose the class (action) with maximum joint probability $p(\boldsymbol{x}, \omega) = p(\boldsymbol{x} \mid \omega)\pi(\omega)$. Assume that the observations from class ω_i are multivariate Gaussian distributed with class-conditional mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$, $N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. Then the log joint probability can be written:

$$\log(p(\boldsymbol{x} \mid \omega_i)\pi(\omega_i)) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i) - \frac{D}{2}\log 2\pi - \frac{1}{2}\log \mid \boldsymbol{\Sigma}_i \mid + \log \pi(\omega_i) \quad (13)$$

where $\mid \cdot \mid$ denotes the determinant and D is the dimension of the input vector. When the covariance matrix for all classes are identical, $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$, rewriting (13) and removing terms independent of i, leads to the linear discriminant function for class ω_i.

$$g_i(\boldsymbol{x}) = \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i + \log \pi(\omega_i) = \boldsymbol{a}_i^T \boldsymbol{x} + a_{i0} \quad (14)$$

Observations are classified to the class with maximum value for the linear discriminant function. The resulting decision boundaries between the classes are hyperplans.

If the covariance matrices are different for each class this leads to quadratic discriminant functions. An early application of the Gaussian based quadratic discriminant function was given by Smith (1947) for $M = 2$ classes. Rewriting (13) and removing one term independent of i, leads to the quadratic discriminant function for class ω_i.

$$\begin{aligned} g_i(\boldsymbol{x}) &= -\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{x} + \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i - \frac{1}{2}\log \mid \boldsymbol{\Sigma}_i \mid + \log \pi(\omega_i) \\ &= \boldsymbol{x}^T \boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{a}_i^T \boldsymbol{x} + a_{i0} \end{aligned} \quad (15)$$

Observations are classified to the class with maximum value for the quadratic discriminant function. The decision surfaces are hyperquadratics. However, the parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are in most cases unknown. To use either rule, equation (14) or (15), the parameters have to be replaced by estimates (the estimative approach). The estimates might be the ordinary maximum likelihood estimates (in the Gaussian situation) or robust estimates. Decision boundaries for the linear and the quadratic discriminant functions are shown in the synthetic example of section 6.

In 1936 R.A. Fisher developed the classical method of linear discriminant analysis (LDA). His aim was to find a linear combination of the variables which maximized the ratio of its between-class to within-class variance, see for example Johnson and Wichern (1982). In this way the observations are transformed so that they are separated as much as possible. His approach has the advantage that it is suited for visual inspection and graphical description , it "separates" the populations. Let the between-class variance be denoted $\boldsymbol{B} = \sum_{m=1}^{M}(\boldsymbol{\mu}_m - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_m - \bar{\boldsymbol{\mu}})^T$, where $\boldsymbol{\mu}_m$ denotes the mean of class ω_m and $\bar{\boldsymbol{\mu}}$ the overall mean. The within-class variance is assumed equal for all classes and is denoted $\boldsymbol{\Sigma}$ ($\boldsymbol{\Sigma}$ is assumed to have full rank). The linear combination $\boldsymbol{l}^T\boldsymbol{X}$ that maximize $\boldsymbol{l}^T\boldsymbol{B}\boldsymbol{l}/\boldsymbol{l}^T\boldsymbol{\Sigma}\boldsymbol{l}$ under the constraint that $\boldsymbol{l}^T\boldsymbol{\Sigma}\boldsymbol{l} = 1$ is found to be the scaled eigenvectors of $\boldsymbol{\Sigma}^{-1}\boldsymbol{B}$ corresponding to the nonzero eigenvalues of $\boldsymbol{\Sigma}^{-1}\boldsymbol{B}$. The eigenvector corresponding to the largest eigenvalue defines the first discriminant $\boldsymbol{l}_1^T\boldsymbol{X}$. The second linear discriminant $\boldsymbol{l}_2^T\boldsymbol{X}$ is constructed from the eigenvector corresponding to the second largest eigenvalue and so on. (We also have $Cov(\boldsymbol{l}_j^T\boldsymbol{X}, \boldsymbol{l}_i^T\boldsymbol{X}) = 0$ for $i \neq j$

and $Var(\boldsymbol{l}_j^T\boldsymbol{X}) = 1$.) The number of linear discriminants equals the number of nonzero eigenvalues, which is smaller than or equal to $\min(D, M-1)$. Observations are assigned to the class of the nearest (Euclidean distance) class mean in the discriminant space. This equals classification to the nearest Mahalanobis distance population mean in the input space. Again, the parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}$ and the between-class variance $\boldsymbol{B}$ are usually unavailable. Replacing the parameters by estimates from the training set leads to Fisher's sample linear discriminants. Ripley (1994a) advocates the use of robust estimates.

Classification using Fisher's linear discriminant (LDA) is equivalent to using (14) with equal prior probabilities $\pi(\omega_i)$=π. To see this, Johnson and Wichern (1982), as above $\boldsymbol{l}_j^T$ is the jth linear discriminant and eigenvector of $\boldsymbol{\Sigma}^{-1}\boldsymbol{B}$. Let $\boldsymbol{e}_j = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{l}_j$ be a scaled eigenvector of $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{\Sigma}^{-\frac{1}{2}}$ and $\boldsymbol{E}$ be the orthogonal matrix of eigenvectors $\boldsymbol{e}_j$. Using LDA an observation $\boldsymbol{x}$ is assigned to the class of the nearest class mean in the discriminant space.

$$\sum_{j=1}^{D}(\boldsymbol{l}_j^T\boldsymbol{x} - \boldsymbol{l}_j^T\boldsymbol{\mu}_i)^2 = \sum_{j=1}^{D}\boldsymbol{l}_j^T(\boldsymbol{x} - \boldsymbol{\mu}_i)^2$$

$$= (\boldsymbol{x} - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{E}\boldsymbol{E}^T\boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x} - \boldsymbol{\mu}_i) = (\boldsymbol{x} - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)$$

Removing terms independent of i leads to the linear discriminant function $\boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i$ which equals equation (14) when the prior probabilities of the classes are equal.

The linear discriminants can also be found using linear regression. This is the starting point for flexible discriminant analysis (FDA) in section 4.6.

4.2 Logistic and Log-Linear Discrimination

The idea behind logistic discrimination is to assume a parametric form for the ratio of two class-conditional distributions instead of specifying a parametric form of the class-conditional distributions themselves, see chapter 6 of Seber (1984). Denote the distribution (density) for class ω_i by f_i. Choosing the distribution for class ω_M as a reference, suppose that

$$\log\frac{f_i(\boldsymbol{x})}{f_M(\boldsymbol{x})} = \alpha_i + \beta_i^T\boldsymbol{x} \qquad i = 1, ..., M-1 \tag{16}$$

The family of distributions satisfying this relationship includes the Gaussian distributions with equal covariance matrices, where $\alpha_m = \boldsymbol{\mu}_m\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_m - \boldsymbol{\mu}_M\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_M$ and $\boldsymbol{\beta}_m = -2\boldsymbol{\mu}_m^T\boldsymbol{\Sigma}^{-1} + 2\boldsymbol{\mu}_M^T\boldsymbol{\Sigma}^{-1}$. Other distributions satisfying the relationship in equation (16) are listed in section 6.4.1 of Seber (1984).

The posterior probability for class ω_i becomes:

$$p(\omega_i|\boldsymbol{x}) = \frac{\pi_i f_i(\boldsymbol{x})}{p(\boldsymbol{x})} = \frac{\exp(\alpha_i + \log\frac{\pi_i}{\pi_M} + \boldsymbol{\beta}_i^T\boldsymbol{x})}{1 + \sum_{l=1}^{M-1}\exp(\alpha_l + \log\frac{\pi_l}{\pi_M} + \boldsymbol{\beta}_l^T\boldsymbol{x})} = \frac{\exp(\alpha_{i0} + \boldsymbol{\beta}_i^T\boldsymbol{x})}{1 + \sum_{l=1}^{M-1}\exp(\alpha_{l0} + \boldsymbol{\beta}_l^T\boldsymbol{x})} \tag{17}$$

where $\alpha_{l0} = \alpha_l + \log\frac{\pi_l}{\pi_M}$. This model for the posterior probabilities was suggested by Cox (1966) and Day and Kerridge (1967) for two classes and extended to more than two classes by Anderson (1972).

The name *logistic* discrimination comes from the situation with only two classes. Then $p(\omega_1|\boldsymbol{x})$ is the logistic function, see equation (17). For more than two classes this approach can be called *log-linear*. Here the situation with more than two classes is emphasized, and therefore we use the name log-linear classification.

The log-linear classifier classifies to the class with the maximum posterior probability. The parameters of the model can be estimated using maximum likelihood estimation. The likelihood to be maximized is constructed on the basis of the posterior probabilities

$$\prod_{i=1}^{M}\prod_{j=1}^{n} p(\omega_i|\boldsymbol{x}_j)^{z_{ij}} \tag{18}$$

where z_{ij} is an 0/1 encoding identifying the correct class for observation $\boldsymbol{x}_j$ from the training set. Using this likelihood the log-linear models are fitted for each class with respect to the reference distribution for class ω_M.

Linear discriminant functions can be produced for the log-linear discrimination,

$$g_i(\boldsymbol{x}) = \log\frac{p(\omega_i|\boldsymbol{x})}{p(\omega_M|\boldsymbol{x})} = \log\frac{\pi_i}{\pi_M} + \alpha_i + \boldsymbol{\beta}_i^T\boldsymbol{x} = \alpha_{i0} + \boldsymbol{\beta}_i^T\boldsymbol{x} \quad for \quad i = 1, ..., M-1$$

and

$$g_M(\boldsymbol{x}) = 0.$$

An observation $\boldsymbol{x}$ is classified to the class ω_i with the maximum discriminant function $g_i(\boldsymbol{x})$.

Comparing the linear discriminant of equation (14) with the log-linear discriminant function above, we see that both methods produce linear class boundaries. The log-linear discrimination method is however more robust, it is correct for other distributions besides the Gaussian distributions with equal covariance matrices. For comparisons between the linear discriminant function and the log-linear discriminant function see Efron (1975) and chapter 6.4.4 of Seber (1984).

It is possible to generalize the log-linear discrimination to include other relationships between the ratio of two class-conditional distributions than the linear relationship in equation (16),

$$\log\frac{f_i(\boldsymbol{x})}{f_M(\boldsymbol{x})} = \phi_i(\boldsymbol{x}) \qquad i = 1, ..., M-1$$

$$p(\omega_i|\boldsymbol{x}) = \frac{\exp\phi_i(\boldsymbol{x})}{1 + \sum_{l=1}^{M-1}\exp\phi_l(\boldsymbol{x})}$$

where $\phi_i(\boldsymbol{x})$ is a general function (parametric, semi-parametric or nonparametric).

4.3 kth Nearest Neighbour Classification

The $k - NN$ classification method has its origin in a technical report of Evelyn Fix and J.L. Hodges Jr in 1951, reprinted in Dasarathy (1991). The method is appealing in its simplicity. The class of a new observation is decided by a majority vote between its k nearest neighbours among the training set observations (ties are broken arbitrarily). The method is built on the assumption that observations which are close together (in the appropriate metric) will have the same classification (at least the same tendency for the posterior distribution for the classes). The value of k is often chosen to be odd to make ties less likely. In a given situation the best value of k can be found using a validation set. That is, the value of k that produces the lowest misclassification rate on the validation set is chosen. Among the class of $k-NN$ rules $1-NN$ is admissible, and therefore no $k-NN$ rule ($k \neq 1$) will have lower probability of error (risk) against all distributions, Cover and Hart (1967).

The $k-NN$ method can also be found as the MAP classification estimate of the kernel method when the class priors are estimated by the proportions of the classes in the training set. Let ω_i denote class i and $\omega(\boldsymbol{x})$ denote the true class of observation $\boldsymbol{x}$. The posterior distribution is estimated as

$$\hat{p}(\omega_i|\boldsymbol{x}) = \frac{\sum_{\omega(\boldsymbol{x}_l)=\omega_i} K(\boldsymbol{x}-\boldsymbol{x}_l)}{\sum_{\boldsymbol{x}_l} K(\boldsymbol{x}-\boldsymbol{x}_l)}$$

where K is an adaptive kernel that is constant over the nearest k points and zero elsewhere. See for example Silverman (1986) for details on the kernel method.

After being a child of statisticians $k-NN$ was a topic of research mainly in the information theoretic and pattern recognition literature. Cover and Hart (1967) found tight bounds on the large sample $k-NN$ risk $B = \lim_{n\to\infty} B(n)$,

$$B^* \leq B \leq B^*(2 - \frac{M}{M-1}B^*)$$

where B is the Bayes risk defined in (2), B^* is the minimum Bayes risk and M the total number of classes. Topics of interest are found in the collection of articles by Dasarathy (1991), for example choice of metric, variations on $k-NN$ and performance of $k-NN$ on real world data sets.

When using $k-NN$ two problems might emerge, storing large data-sets and rejecting outliers. A method called data-editing can cope with this. Data-editing consists in editing the data and deleting observations from the data set. A method called multiedit is described in Devijver and Kittler (1982).

4.4 Classification and Regression Trees (CART)

CART is a recursive partitioning method to be used both for regression and classification. The main reference is Breiman et al. (1984).

Let $v(\boldsymbol{x})$ be the function we want to approximate (estimate). For classification this might be regarded as the Bayes decision rule classifier; a function giving the class of an observation $\boldsymbol{x}$.

The Algorithm from Friedman (1994)

1. Divide the input space, usually $\mathbb{R}^D$, into disjoint regions $\{R_l\}_{l=1}^K$
2. Approximate the function v with a constant inside each region,

$$v(\boldsymbol{x}) = a_l \qquad for \quad \boldsymbol{x} \in R_l.$$

The model is equivalent to a basis function expansion,

$$v(\boldsymbol{x}) = \sum_{l=1}^{K} a_l I(\boldsymbol{x} \in R_l)$$

where I is an indicator function.

A recursive search strategy is used to divide the input space into disjoint regions. At each step one input variable or linear combination of input variables is split to create two new basis functions, deleting the term that was split on. This is done until no more splits are viable, determined by a measure of impurity. A merging step concludes the recursive strategy. After a tree is constructed it can be pruned to achieve better generalization.

The resulting model can be represented as a binary tree. This makes the results from using CART easy to visualize. In addition to being a good tool for classification, CART is also valuable as an explanatory tool and for feature extraction.

4.5 Using Methods from Regression to Construct Discriminant Functions

In section 2.3 the construction of a classifier was based on a set of discriminant functions, the map $\delta : \mathcal{X} \to \mathcal{Y}$ was the composition $\delta(\boldsymbol{x}) = h(\boldsymbol{g}(\boldsymbol{x}))$. The MLP in section 3 gave a parametric choice for the discriminant function $\boldsymbol{g}$ (given the number of nodes). Here the form of $\boldsymbol{g}$ is described by regression methods.

There are several ways of presenting the class of an observation as a response random variable. One possibility is to regard the class as a polytomous nominal response random variable. Here we use the definition from section 2.3 and let t_i be a dichotomous random variable denoting class ω_i or not for $i = 1, ..., M$, see equation (5). Then each class is encoded as a target $\boldsymbol{t}$.

Two possible choices for the relationship between the discriminant function and the posterior class probabilities are presented,

$$E(t_i|\boldsymbol{x}) = p(\omega_i|\boldsymbol{x}) = g_i(\boldsymbol{x}) \qquad i = 1, ..., M$$

and

$$E(t_i|\boldsymbol{x}) = p(\omega_i|\boldsymbol{x}) = \frac{\exp g_i(\boldsymbol{x})}{1 + \sum_{l=1}^{M-1} \exp g_l(\boldsymbol{x})} \qquad i = 1, ..., M.$$

In the first relationship the discriminant function and the posterior class probability distribution are equal. The second relationship is a generalization of the log-linear discrimination in section 4.2.

The classification problem can be defined as a regression problem; regress the input variable $\boldsymbol{x}$ onto the target $\boldsymbol{t}$ to optimize a chosen error measure. Both the form of $\boldsymbol{g}$ (parametric, semiparametric or nonparametric) and the choice of loss function L_h^* depend on the regression method in question. The mapping h is to choose the class corresponding to the largest value in $\boldsymbol{g}$.

Here generalized additive models (GAM), projection pursuit regression and multivariate adaptive regression splines (MARS) are presented briefly.

4.5.1 Generalized Additive Models (GAM)

The term *additive model* states that a model is written as a sum of terms, some terms may be nonparametric, others may be linear or functions of more than one predictor variable. The additive model can be regarded as a generalization of the linear model. An additive model for the mean of the ith response variable can be presented as

$$g_i(\boldsymbol{x}) = \alpha_i + \sum_{k=1}^{D} f_{ik}(x_k)$$

where α_i are constant coefficients and f_{ik} are smooth univariate functions. The function $g_i(\boldsymbol{x})$ is the mean of the response random variable, $g_i(\boldsymbol{x}) = E(t_i|\boldsymbol{x})$. The function f_{ik} is called a smoother. A smoother is a tool for summarizing the trend of a response measurement. The smoother produces an estimate of the trend that is less variable than the response measurement itself. Most smoothers attempt to mimic averaging over intervals (or categories) through local averaging.

The generalized additive model (GAM) is presented in Hastie and Tibshirani (1990). Here the distribution of the response and the link between the distribution of the response and the predictors can take quite *general* forms, a *generalized* model.

As an example of a generalized additive model let t_i be a dichotomous random variable Bernoulli distributed with conditional mean $\boldsymbol{\mu}_i = E(t_i|\boldsymbol{x})$, where t_i is given in equation (5). The mean can be linked to the additive model of the predictors via a link function $l(\boldsymbol{\mu}_i)$,

$$l(\boldsymbol{\mu}_i) = \alpha_i + \sum_{k=1}^{D} f_{ik}(\boldsymbol{x}_k).$$

Possible choices for the link function are for example the logit, probit, log-log and complementary log-log function, see McCullagh and Nelder (1989). The logit function (the canonical link in this situation) is

$$\begin{aligned} l(\boldsymbol{\mu}_i) &= \operatorname{logit}(\boldsymbol{\mu}_i) = \log \frac{\boldsymbol{\mu}_i}{1-\boldsymbol{\mu}_i} \\ &= \log \frac{P(t_i = 1|\boldsymbol{x})}{P(t_i = 0|\boldsymbol{x})} = \log \frac{p(\omega_i|\boldsymbol{x})}{1 - p(\omega_i|\boldsymbol{x})} = \alpha_i + \sum_{k=1}^{D} f_{ik}(\boldsymbol{x}_k). \end{aligned}$$

This is a generalized additive model for class ω_i (separately) versus all other classes. For our classification problem M separate additive logit models are necessary. This means fitting separate logistic models for each class versus the rest without imposing consistency conditions on the posterior probabilities. To produce posterior probabilities from the logit GAM,

$$p(\omega_i|\boldsymbol{x}) = \frac{\exp(\operatorname{logit}(\boldsymbol{\mu}_i))}{1+\exp(\operatorname{logit}(\boldsymbol{\mu}_i))}. \tag{19}$$

(We see that this is of the same form as fitting a MLP with logistic output activation function and different activation functions for each node in the hidden layer for each output node.) To assure that the posterior probabilities from M separate logit GAMs sum to one, one may use the following scaling,

$$p(\omega_i|\boldsymbol{x}) = \frac{[\exp(\operatorname{logit}(\boldsymbol{\mu}_i))/(1+\exp(\operatorname{logit}(\boldsymbol{\mu}_i)))]}{\sum_{j=1}^{M}[\exp(\operatorname{logit}(\boldsymbol{\mu}_j))/(1+\exp(\operatorname{logit}(\boldsymbol{\mu}_j)))]}.$$

Using the Bayes decision rule with zero-one loss function results in choosing the class ω_i with the largest posterior probability $p(\omega_i|\boldsymbol{x})$. Since the (logit) odds transform $p(\omega_i|\boldsymbol{x})/(1-p(\omega_i|\boldsymbol{x}))$ is a monotone one-to-one function, choosing the class with the largest posterior probability $p(\omega_i|\boldsymbol{x})$ will be the same as choosing the class with the largest odds ratio. This means that the scaling

$$\frac{\exp(\mathrm{logit}(\boldsymbol{\mu}_i))}{\sum_{j=1}^{M}\exp(\mathrm{logit}(\boldsymbol{\mu}_j))}$$

will give the same classification as equation (19).

A generalized additive model can also be seen as a generalization of a generalized linear model (GLM); replacing the linear terms by additive ones. A GLM can be fitted using the iteratively-reweighted least-squares procedure for computing the maximum likelihood estimates in a GLM, McCullagh and Nelder (1989). A generalized additive model (GAM) is fitted by a method called local scoring, Hastie and Tibshirani (1990).

Model selection techniques for GAM choose which terms to include in the model and also how smooth the terms should be. The algorithm BRUTO combines the method of backfitting and the selection of a smoothing parameter by minimizing a statistic similar to generalized cross-validation, see Hastie and Tibshirani (1990).

4.5.2 Projection Pursuit Regression

The projection pursuit regression method PPR of Friedman and Stuetzle (1981) models a regression surface of one response variable as a sum of general smooth nonlinear functions of linear combinations of predictor variables. The algorithm approximates a general function of D variables by a sum of nonlinear functions of linear combinations of the predictors,

$$g(\boldsymbol{x}) = \sum_{k=1}^{K} \phi_k(\boldsymbol{\beta}_k^T \boldsymbol{x}).$$

Fitting a PPR model starts by choosing a direction $\boldsymbol{\beta}$ and a smooth function ϕ that minimize the square average of the residuals $r = y - g(\boldsymbol{x})$ for the observations $\boldsymbol{x}$. Here y denotes the (univariate) response measurement. This process, denoted backfitting, is repeated iteratively, using the residuals r in the place of the response y, Friedman and Stuetzle (1981).

Diaconis and Shahshahani (1984) discuss necessary and sufficient conditions for approximating a given function g when the functions ϕ_k are univariate, nonlinear functions. This additive model has the ability to model interactions of the predictor variables. Because of the use of linear combinations there is only need for univariate smoothers. See Huber (1985) for an overview of projection pursuit.

SMART (Smooth Multiple Additive Regression Technique) of Friedman (1984), is a generalization of projection pursuit regression to multiple response regression and classification. SMART models a set of response variables as a function of a set of predictor variables. The model for the ith response can be written

$$E(t_i|\boldsymbol{x}) = g_i(\boldsymbol{x}) = \eta_i + \sum_{k=1}^{K} \gamma_{ik}\phi_k(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}).$$

The same set of basis functions ϕ_k and direction parameters α_k and $\boldsymbol{\beta}_k$ are used for all the responses. A further generalization is also possible. Response specific basis functions $\phi_k = \phi_{ik}$ can be used and also response specific parameters $\alpha_k = \alpha_{ik}$ and $\boldsymbol{\beta}_k = \boldsymbol{\beta}_{ik}$. In the SMART model the coefficients γ_{ik} and α_k and the basis functions ϕ_k are estimated by (weighted) least squares. This is different from PPR which chooses α_k in a forward stagewise manner. SMART treats classification as a multi-response regression method with the indicator variable encoding (5) of the classes as the responses. Two different loss functions are suggested, a weighted quadratic form in $(t_i - E(t_i|\boldsymbol{x}))$ and the minimum of a weighted sum of $E(t_i|\boldsymbol{x})$ over the different classes. The first choice is recommended because of potential problems with nonconvexity of the second choice, see Friedman (1984).

Comparing PPR/SMART with MLP we see that PPR/SMART consists of a few functions of linear combinations while MLP consists of many function of many linear combinations. The functions of PPR/SMART are adapted to the form of the data while the functions in MLP are not influenced by the data (non-adaptive).

4.5.3 Multivariate Adaptive Regression Splines (MARS)

Multivariate adaptive regression splines of Friedman (1991) can be viewed as a generalization of CART. The algorithm below presents MARS in the same form as CART (see 4.4).

Let $v(\boldsymbol{x})$ be the function we want to approximate (estimate). For classification this might be the function $v(\boldsymbol{x}) = E(t_i|\boldsymbol{x})$.

The Algorithm Friedman (1994)

1. Divide the input space, usually $\mathbb{R}^D$, into disjoint regions $\{R_l\}_{l=1}^{K}$
2. Approximate the function v with a qth degree polynomial inside each region,
$$v(\boldsymbol{x}) = P_l^q \qquad for \quad \boldsymbol{x} \in R_l.$$
3. The choice of functions are constrained so that $\frac{\partial^k v}{(\partial \boldsymbol{x})^k}$ is continuous for $k = 0, ..., q-1$.

The goal is to use the data to simultaneously estimate a good set of subregions and the parameters associated with the separate functions in each subregion. Continuity at subregion boundaries is enforced. CART does not have continuity at subregion boundaries.

The MARS model can be written in terms of adaptive regression splines,

$$v(\boldsymbol{x}) = \alpha_0 + \sum_{l=1}^{L} \alpha_l B_l(\boldsymbol{x}) = \alpha_0 + \sum_{l=1}^{L} \alpha_l \prod_{k=1}^{K_l} [s_{kl}(\boldsymbol{x}_{\nu(k,l)} - y_{kl})]_+$$

where K_l is the number of splits that gave rise to the lth basis function B_l. The largest value of K_l is the *degree* of the model (compared to q in the algorithm above). The scalar s_{km} is ± 1. The set $\nu(k,l)$ label the predictor variables and the splits y_{kl} represents values on the corresponding variables.

MARS consists of a forward stepwise and a backward stepwise algorithm. A MARS model is fitted using a weighted least squares loss function, that is, a modified generalized cross-validation criterion. MARS is reported to do well for moderate sample sizes; $50 \leq n \leq 1000$, and moderate to high dimensions; $3 \leq D \leq 20$, see Friedman (1991).

The above MARS model is constructed for one response variable. When MARS is used as a method for classification, M separate MARS models are fitted to the M response variables t_i, $i = 1, ..., M$.

4.6 Flexible Discriminant Analysis (FDA)

Flexible discriminant analysis is a method for discrimination and classification combining the powers of multi-response nonparametric regression techniques and optimal scoring, Hastie et al. (1994). The same ideas have been independently taken up by Ripley (1994b).

The motivation for FDA lies in the correspondence between linear discriminant analysis, canonical correlation analysis and optimal scoring. Linear discriminant analysis can be performed as a multi-response linear regression using optimal scoring to represent the classes.

In section 4.1 it is shown that the sampling-paradigm classifier for class-conditional Gaussian distributions with equal covariance matrices, can be written as a linear discriminant function. For two classes the sample linear discriminant can be found using linear regression of the class indicators (target) of one of the classes on the predictors, see Anderson (1984) or Ripley (1993b). The situation is generalized to more than two classes in unpublished work by Breiman and Ihaka (1984).

For more than two classes the target $\boldsymbol{t}$ is replaced by a $M-1$ dimensional score $\boldsymbol{\theta}(\boldsymbol{t})$. More general, the linear discriminant classifier can be found regressing the scores $\boldsymbol{\theta}(\boldsymbol{t})$ on the predictors using least squares estimation of *both* the scores and the parameters in the regression. In Hastie et al. (1994) it is shown that this can be done in separate steps for the scores and the parameters. First initial scores are set to the 0/1 indicator variable encoding of the classes. The encodings for all n observations in the training set form the $n \times M$ matrix $\boldsymbol{T}$. A linear multi-response multivariate regression is fitted to give a matrix of estimated parameter coefficients $\hat{\boldsymbol{B}}$, yielding fitted values $\hat{\boldsymbol{T}} = \boldsymbol{P_x T} = \boldsymbol{X}\hat{\boldsymbol{B}}$. Optimal scores $\boldsymbol{\Theta}$ are found as the eigenvector matrix of $\boldsymbol{T}^T\hat{\boldsymbol{T}}$ with the normalization $\boldsymbol{\Theta}^T \frac{1}{n}\boldsymbol{T}^T\boldsymbol{T}\boldsymbol{\Theta} = \boldsymbol{I}$. As a final step the parameter coefficient matrix $\hat{\boldsymbol{B}}$ is updated to reflect that the regression is with responses $\boldsymbol{T\Theta}$ rather than $\boldsymbol{T}$, $\hat{\boldsymbol{B}} = \boldsymbol{P_x T \Theta}$. Since the regression is linear in $\boldsymbol{T}$ the optimization over scores can be done without refitting the regression.

The FDA procedure is equivalent to performing a linear discriminant analysis in the space of fitted values from the regression of the class indicators on the predictors, Hastie et al. (1994) and Ripley (1994b). Flexible discriminant analysis replaces the linear regression step by a nonparametric or semi-parametric regression. This opens for a variety of regression tools to yield different discrimination rules and flexible class boundaries. See Hastie et al. (1994) for further details and examples.

Ripley (1994a) regards this flexible regression step as choosing a transformation of the sample space in which linear discriminant analysis is used. This means that the above philosophy can be adapted to most statistical software. Ripley (1994b) and Ripley (1994a) give examples where it is not adequate to perform LDA in the transformed space.

5 The Marine Phytoplankton Data Set

The marine phytoplankton data set consists of spectral light absorption characteristics of 31 species of bloom-forming phytoplankton from Johnsen et al. (1994). In their study Johnsen et al. (1994) conclude that different phytoplankton classes may be identified during blooms on the basis of absorption measurements of 3 to 5 appropriately chosen wavelengths. The method of identification is linear discriminant analysis (LDA). Here we use flexible discriminant analysis (FDA) and MLP's on the marine phytoplankton data set to compare the performance of these methods.

In Johnsen et al. (1994) the in vivo absorption spectra were log-transformed, normalized and discretized. Outliers were removed using biological reasoning. The final data set consists of 217 observations of 85 wavelength variables and three environment variables (day-length, temperature and light condition under growing). The number of observations of each species varies from 2 to 54. This variation does not necessarily reflect the size of the population of each species found in nature and should therefore be used with caution as an indication of the prior probabilities for the species.

For the classification method to be used in practice, it is not economically possible to base the classification rule on all 85 wavelength variables. Furthermore, the 85 wavelengths variables are strongly correlated. From the 85 correlated variables a small number of variables that distinguish best between the different species must be found. In Johnsen et al. (1994) three input variables were selected using stepwise linear discriminant analysis. As a screening, a slightly different approach involving hierarchical clustering and linear discriminant analysis was tried here. The 85 variables were divided into six groups using hierarchical clustering (average criterion). Inside each cluster the variables were strongly correlated (minimum correlation of 0.26 to 0.90 inside each cluster). The six clusters were represented by six variables, choosing for each cluster the variable with the highest correlation to all the other variables in this cluster. The resulting six variables had low correlation among themselves. The 20 possible selections of three variables from these six was investigated using LDA. The resulting three best wavelengths (using one training set and one test set) were highly correlated with the choice in Johnsen et al. (1994). Because of this and to yield a common platform for comparison, the three input variables from Johnsen et al. (1994) (absorptions values at 481, 535 and 649 nm) are used in all the analyses here. One of the environment variables, the value for light condition, was found to be of importance. On the basis of the light condition the data set can be divided into a low-light and a high-light data set. Four different settings were analysed:

- Total data set with the three wavelength inputs (3 inputs, 31 species and 217 observations).
- Total data set with the three wavelength inputs and one continuous input for scalar light condition (4 inputs, 31 species and 217 observations).
- High-light data set with three wavelength inputs (3 inputs, 25 species and 69 observations).
- Low-light data set with three wavelength inputs (3 inputs, 28 species and 145 observations).

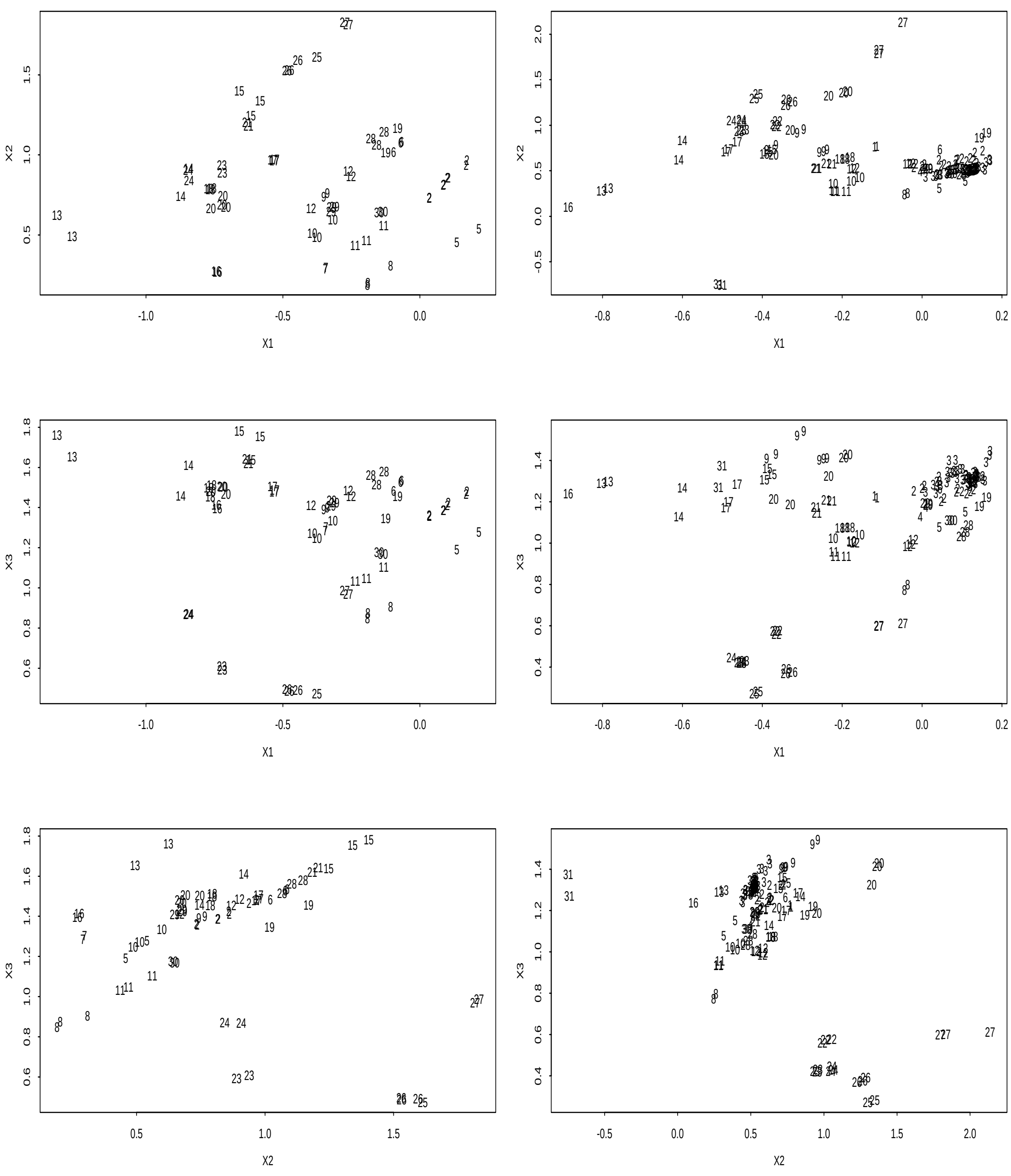

Figure 2: The phytoplankton data, high-light data in the left column and low-light data in the right column. X1, X2 and X3 denote the three wavelength variables and the labels 1 to 31 denotes the 31 species.

5.1 Analyses with MLP's

The multilayer perceptron (MLP) is presented in section 3.

Architecture

A multilayer perceptron with one hidden layer was used. Hidden and output layers had sigmoid activation function with output in $[-\frac{1}{2},\frac{1}{2}]$. The number of output nodes (M) was equal to the number of species, and their target values were $-\frac{1}{2}$ (off) and $\frac{1}{2}$ (on) ($\mathcal{T} = \{-\frac{1}{2}, \frac{1}{2}\}^M$).

The number of input and output nodes was determined by the four settings (number of inputs equals three or four and number of outputs equals the number of species present). Different numbers of hidden nodes were tried, $H \in \{0, 1, ..., 25\}$.

Learning

The analyses were done using the software Aspirin/MIGRAINES 6.0 from the Mitre Corporation, Leighton (1992). The learning method was back-propagation with momentum. Euclidean loss function was used. A pattern was classified to the class with maximum value for the output node, see (6).

Unless specific measures are taken Aspirin/MIGRAINES presents the training samples in a given sequential (not random) order and sequential presentation was used here. For these data sets updating after each training pattern introduced too much noise into the training procedure, while batch update tended to find and get stuck in local minima. Conjugate gradient search was quickly trapped in a local minimum. Updating after every 10th pattern gave a substantial improvement in convergence speed.

The momentum term was in all analyses set at 0.95. The learning rate was $\eta = 0.1$ for the total data sets and $\eta = 0.01$ for the divided data sets. (A too large learning rate is easily detected if the average squared residual (ASR) on the training set is plotted against time, it gives spike-like changes.)

Parameter Selection and Network Evaluation Criteria

After deciding to use a MLP with one hidden layer, sigmoid activation function, back-propagation learning method, quadratic loss function, maximum output activation classification criterion, update interval, learning rate and momentum, two problems remain:

1. Selection of the remaining network architecture and learning parameters (the number of hidden nodes H and the stopping point for learning).

2. Evaluation of network performance. How accurate is the network for predicting future observations?

To find the best number of hidden nodes each of the data set were divided into one training set and one test set. This is due to limited time and computational resources. These test sets did not necessarily reflect the distribution of classes in the training set.

To find the best number of hidden nodes different networks were trained for approximately 5000 epochs each. All networks started with initial configuration for the network weights randomly chosen from a uniform distribution on $[-0.10, 0.10)$. Aspirin/MIGRAINES also add uniform noise [0,0.05] to the calculation of sigmoid derivatives in the back-propagation algorithm. This is to improve the speed of convergence. The different network architectures were evaluated using the ASR, see equation (9), and percent of wrongly classified cases (error rate) on the training set and on the test set.

After the best network architecture was chosen, this network was trained for at least 10 000 epochs to find a good point to stop training. After each 250 epoch the error on the test set was calculated. The ASR was scaled with the number of output nodes in the network to allow comparison between the different settings. In general, to avoid overfitting, training should be stopped before the ASR on the training set is "too" small. Here training was stopped when the ASR on the test set had a minimum and the number of wrongly classified observation in the test set was small. This will of cause depend on the chosen test set and the stopping point for training might be different for different test sets. An alternative method would be to examine the ASR curve on the training set. This curve often has a breakpoint before becoming relatively horizontal. Training can be stopped at this breakpoint (if the breakpoint can be easily identified). The reason for using both ASR and percent of wrongly classified observations is that a decrease in ASR does not necessarily mean a decrease in percent wrongly classified observations and the percent of wrongly classified observation is therefore used to supply additional information.

To get a good estimate of the performance of the optimal network v-fold cross-validation was used (11-fold for the total data sets and 7-fold for the divided data sets), see section 2.5, to give the ASR and the percent wrongly classified observations. The network weights were given different initial values for each training/test set pair, and the network weights were estimated once for each training/test pair.

Due to the large amount of parameters to be selected, a number of decisions have been made on the basis of trial (learning rate, update interval and so on). Only the selection of the number of hidden nodes and the stopping point is reported. The fact that one training set and one test set was used to select the network architecture and the stopping point for training will slightly bias the further comparison of the MLP with other methods for discrimination and classification. A better solution would have been to use cross-validation, but this was found to be too time consuming. Another potential weakness in this analysis is that the number of weights used is larger than the number of training samples and could cause overfitting, but since early stopping is used this problem is overcome because the effective number of parameters (weights) are substantially lower than the actual number of parameters (weights), see Moody (1992) and Sjöberg (1993).

Results for 3-H-31 Networks, Total Data Set

For the $3 - H - 31$ networks (3 nodes in the input layer, H nodes in the hidden layer and 31 nodes in the output layer) the input variables were the three selected wavelengths. 17 different networks were trained for approximately 5000 epochs to find the optimal number of hidden nodes, H. The results are given in table 7 (in the appendix) over ASR and the percent of observations classified wrongly for the training and the test set.

Classification of the Total Marine Phytoplankton Data Set with 3 Explanatory Variables Using a 3-15-31 MLP Network				
	mean	sd.dev	min	max
ASR on test sets	0.00650133	0.00235034	0.00280007	0.0107562
ASR on training sets	0.000278151	7.75954e-05	0.000179462	0.000433307
% wrong on test sets	9.76	5.64016	0.00	15.79
% wrong on training sets	0.51	0.374889	0.00	1.52

Table 1: Classification of the total marine phytoplankton data set with 3 explanatory variables and 31 species using a MLP with 15 hidden nodes, a 3-15-31 network. The results of the table is from 11-fold cross-validation. The columns give the mean, standard deviation, the minimum and the maximum of the 11 cross-validation runs for the ASR and the error rates (%) for the training sets and the test sets. The total data set has 217 observations, which give training sets of size 197-198 and test sets of size 19-20.

From the table we see that at fixed 5000 epochs a large network fits the training data better than smaller networks. The two largest networks would also give better fits if they were trained for some epochs more. Minimum value for the ASR on the test set was in most cases obtained near 5000 epochs, see table 8 in the appendix.

Based on the performance on the test set, the best network was chosen to have 15 hidden units, a 3-15-31 network. This network was trained for 14 000 epochs to find the optimal point to stop training. This point is where the ASR for the test set has a minimum and the percent of wrongly classified observations is low, see figure 6. The optimal point was found at 7800 epochs (approximately 31 passes of 250 epochs).

The results for 11-fold cross-validation on the 3-15-31 network trained for 7800 epochs are found in table 1. The columns give the average, standard deviation, minimum and maximum over the 11 cross-validation runs.

Aspirin/MIGRAINES does not update the weights in the network if the average error is smaller than a predefined constant (0.00005). During learning only 5500 of the 7800 epochs had been subject to weight change. This is an indication that the network has learned.

Results for 4-H-31 Networks, Total Data Set

For the $4 - H - 31$ networks the input variables were the three selected wavelengths and the scalar light condition (scaled). 14 different networks were trained for approximately 5000 epochs to find the optimal number of hidden nodes, H. The results are given in table 9 (in the appendix) over ASR and the percent of observations classified wrongly for the training and the test set.

The network 4-13-31 was chosen since it had the lowest ASR on the test set. This network was trained for 14 000 epochs to find the optimal point to stop training. This point is where the ASR for the test set has a minimum and the percent of wrongly classified observations is low, see figure 7. The optimal point was found at 5000 epochs (20 passes of 250 epochs). Since the 4-13-31 network has fewer weights to be estimated than the 3-15-31 network, it is intuitively plausible that training can be stopped earlier here.

Classification of the Total Marine Phytoplankton Data Set with 4 Explanatory Variables Using a 4-13-31 MLP Network				
	mean	sd.dev	min	max
ASR on test sets	0.00672498	0.00924445	0.000581628	0.0336179
ASR on training sets	0.00115896	0.000282512	0.000609824	0.00152753
% wrong on test sets	9.76	5.68351	0.00	21.05
% wrong on training sets	1.47	0.847998	0.00	3.03

Table 2: Classification of the total marine phytoplankton data set with 4 explanatory variables and 31 species using a MLP with 13 hidden nodes, a 4-13-31 network. The results of the table is from 11-fold cross-validation. The columns give the mean, standard deviation, the minimum and the maximum of the 11 cross-validation runs for the ASR and the error rates (%) for the training sets and the test sets. The total data set has 217 observations, which give training sets of size 197-198 and test sets of size 19-20.

The results for 11-fold cross-validation on the 4-13-31 network trained for 5000 epochs are given in table 2. The columns give the average, standard deviation, minimum and maximum over the 11 cross-validation runs.

Results for 3-H-25 Networks, High-light Data Set

Linear discriminant analysis gave very good results for this setting in Johnsen et al. (1994). Therefore, it is reasonable to assume that the high-light data set is strongly linear and therefore here small networks were investigated. A total of 19 different network architectures were trained for 10 000 epochs. The results are found in table 10 (in the appendix) .

The best network was chosen to be 3-7-25. For this small high-light data set the network converged faster updating after every fifth pattern. Using minimum ASR on the test set as the stop criterion training was stopped after 17000 epochs. This gave very bad results with 7-fold cross-validation. Training was instead stopped at minimum percent of misclassification on the test set. This was after 5000 epochs. The results using 7-fold cross-validation are given in table 3. The columns give the average, standard deviation, minimum and maximum over the 7 cross-validation runs.

Results for 3-H-28 Networks, Low-light Data Set

Also here the good results with linear discriminant analysis from Johnsen et al. (1994) lead to the conclusion that there must be a strong linear component in the low-light data set. A total of 19 different network architectures were trained for 5000 epochs, results are found in table 11 (in the appendix).

The network chosen to be the best was $3-11-28$. For the low-light data set faster convergence was obtained updating after every seventh pattern. Using minimum ASR on the test set as the stop criterion training was stopped after 6000 epochs. The results using 7-fold cross-validation are found in table 4. The columns give the average, standard deviation, minimum and maximum over the 7 cross-validation runs.

Classification of the Marine Phytoplankton High-Light Data Set with 3 Explanatory Variables Using a 3-7-25 MLP Network				
	mean	sd.dev	min	max
ASR on test sets	0.006177	0.004893	0.002519	0.017882
ASR on training sets	0.001361	0.000680	0.000421	0.002290
% wrong on test sets	22.22	11.8783	11.11	44.44
% wrong on training sets	0.48	0.752938	0.00	1.67

Table 3: Classification of the marine phytoplankton high-light data set with 3 explanatory variables and 25 species using a MLP with 7 hidden nodes, a 3-7-25 network. The results of the table is from 7-fold cross-validation. The columns give the mean, standard deviation, the minimum and the maximum of the 7 cross-validation runs for the ASR and the error rates (%) for the training sets and the test sets. The total data set has 69 observations, which give training sets of size 59-60 and test sets of size 9-10.

Classification of the Marine Phytoplankton Low-Light Data Set with 3 Explanatory Variables Using a 3-11-28 MLP Network				
	mean	sd.dev	min	max
ASR on test sets	0.0099	0.006467	0.000317	0.018747
ASR on training sets	0.001033	0.000272	0.000512	0.001410
% wrong on test sets	6.9	4.33269	0.00	14.23
% wrong on training sets	0.69	0.67177	0.00	1.61

Table 4: Classification of the marine phytoplankton low-light data set with 3 explanatory variables and 28 species using a MLP with 11 hidden nodes, a 3-11-28 network. The results of the table is from 7-fold cross-validation. The columns give the mean, standard deviation, the minimum and the maximum of the 7 cross-validation runs for the ASR and the error rates (%) for the training sets and the test sets. The total data set has 145 observations, which give training sets of size 124-125 and test sets of size 20-21.

5.2 Analyses with FDA

An extensive analysis using stepwise linear discriminant analysis (LDA) is done in Johnsen et al. (1994). Their best results where obtained dividing the data set into high and low scalar light conditions giving 93% correct classification (leave-one-out cross-validation) on both high- and low-light data sets. The results on the divided data sets are not overcome using any other method investigated here.

The analyses with FDA was done using the `fda` library of T. Hastie and R. Tibshirani available from the statistics archives at Carnegie-Mellon University (`statlib@lib.stat.cmu.edu`). The results for flexible discriminant analysis using LDA, MARS (degree=1), MARS (degree=2) and BRUTO are found in table 5. For the high-light data set FDA/BRUTO was not able to discriminate. To make possible comparison with the results found using MLP's, 11-fold and 7-fold cross-validation were used for the total data sets and the divided data sets, respectively.

5.3 Conclusions

Using flexible methods for classification we have been able to improve substantially upon the results of Johnsen et al. (1994) on the total data set with 3 inputs. For this data set the 3-15-31 MLP gave 9.76% wrongly classified (11-fold cross-validation) compared to 39.11% for LDA, see table 5. Adding the extra environment input variable gave no improvement for the MLP, but slight improvements for the LDA and FDA. However, it is clear that the environment variable scalar light condition has a large influence. The total data sets did not seem to be linearly separable. The classification on the total data sets was improved by approximately 10% using FDA/MARS without interactions instead of LDA, see section 4.5.3. Adding two and three term interactions did not give substantial improvement. An explanation for this can be that the input variables presents absorption values at different wavelengths and are not uncorrelated, therefore including interaction terms has little effect.

The divided data sets seem to be nearly linearly separable and inclusion of nonlinear terms or interaction terms did not improve the classification. Using MLP on the divided data sets good results are obtained for the low-light data set, but for the high-light data set a network and an optimal stop point that does not lead to overfitting has not been found.

The FDA procedure is a much faster method than MLP. Finding the best MLP architecture and the optimal training period is a tedious procedure. Having decided the MLP architecture and the activation functions, MLP is a parametric model. FDA is able to use nonparametric regression techniques and is therefore nonparametric in nature. Nevertheless, from the empirical results found here it may seem that the MLP's are able to fit the total data sets (with 3 and with 4 inputs) better than FDA. MLP will always give results (possibly a poor result), while FDA might not be able to discriminate. Both the MLP and the FDA methods use the class frequencies in the training set as an estimate for the class priors.

When data is scarce, the simplest model often give the best results. This might be the case here for the divided data sets. More flexible models do easily overfit the data. Larger data sets benefit more from flexible models. The smallest divided data set with 69 observations is overfitted with the MLP model used.

The final conclusion for the data set of marine phytoplankton is to classify observations with

Cross-Validation Results for the Marine Phytoplankton Data Set									
Method	crit.	Total, 3 inputs		Total, 4 inputs		High-light		Low-light	
		% wrong on		% wrong on		% wrong on		% wrong on	
		train	test	train	test	train	test	train	test
MLP	mean	0.51	9.76	1.47	9.76	0.48	22.22	0.69	6.90
	sd.dev	0.11	1.70	0.26	1.71	0.28	4.49	0.25	1.64
FDA/LDA	mean	29.58	39.11	21.98	34.52	1.21	7.30	1.73	3.44
	sd.dev	0.54	2.15	0.71	2.05	0.31	2.89	0.28	2.01
FDA/MARS (1)	mean	18.25	30.41	12.72	24.88	0.48	10.16	3.68	9.59
	sd.dev	0.45	2.80	1.27	1.04	0.31	3.09	0.42	2.52
FDA/MARS (2)	mean	17.28	28.59	12.21	22.11	0.00	11.59	15.55	19.08
	sd.dev	0.47	2.38	0.99	2.04	0.00	4.03	13.81	13.62
FDA/BRUTO	mean	26.77	36.79	15.85	26.63	-	-	2.87	7.59
	sd.dev	0.73	2.02	0.70	2.64	-	-	0.39	2.04

Table 5: This is a summary of the v-fold cross-validation results for all four marine phytoplankton data sets using multilayer perceptrons and flexible discriminant analysis. The "Total 3, inputs" data set with 3 explanatory variables has 217 observations and 11-fold cross-validation is used. The "Total, 4 inputs" data set with 4 explanatory variables has also 217 observations and 11-fold cross-validation is used. The "High-light" data set has 3 explanatory variables, 69 observations and 7-fold cross-validation is used. The "Low-light" data set has 3 explanatory variables, 145 observations and 7-fold cross-validation is used. The column denoted "train" give the error rates (means and standard deviation of the means) on the training sets and "test" on the test sets. The rows of the table denoted "mean" give the mean of the cross-validation runs and the rows denoted "sd.dev" give the standard deviation of the mean. The MLP has an architecture of 3-15-31 for the total data set with 3 inputs, 4-13-31 for the total data set with 4 inputs, 3-7-25 for the high-light data set and 3-11-28 for the low-light data set. The FDA/MARS (1) has interactions of degree 1 and FDA/MARS (2) of degree 2. For the FDA/BRUTO no result was obtained for the high-light data set. Based on the means and standard deviations of the means on the test sets we can perform tests for comparing the MLP and FDA methods. All tests are based on an approximate 5 % significance level. For the total data set with 3 inputs the MLP is significantly better than each of the other methods. The FDA/MARS (2) is significantly better than the LDA. Also for the total data set with 4 inputs the MLP is significantly better than each of the other methods. Both the FDA/MARS methods are significantly better than the LDA. For the high-light data set the LDA is significantly better than the MLP. There was no significant difference between any two methods for the low-light data set. We will conclude the comparisons by pointing out that the results for the MLP are slightly biased because the model selection (network architecture and stopping point for training) was based on dividing the data set into one training set and one test set.

unknown light conditions by using the MLP solution with the three wavelength inputs. For observation with known and stable light conditions using LDA separately for the high-light and low-light data is preferred

6 The Three Class Synthetic Data Set

In the three class synthetic data set each class is simulated from an equal mixture of two two-dimensional Gaussian distributions. The training set consists of 600 observations, 100 from each component density. The test set and the validation set are also of size 600 observation. The marginal distribution can be written:

$$p(\boldsymbol{x}) = \sum_{i=1}^{3} \frac{1}{3} \sum_{l=1}^{2} \frac{1}{2} f_{il}(\boldsymbol{x}|\boldsymbol{\theta}_{il})$$

where $f_{il}(\boldsymbol{x}|\boldsymbol{\theta}_{il})$ denotes the two-dimensional Gaussian distribution with parameters $\boldsymbol{\theta}_{il} = (\boldsymbol{\mu}_{il}, \boldsymbol{\Sigma}_i)$.

$$\boldsymbol{\mu}_{11} = \begin{bmatrix} 0 \\ -2 \end{bmatrix} \qquad \boldsymbol{\mu}_{12} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \qquad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.8 \end{bmatrix}$$

$$\boldsymbol{\mu}_{21} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \qquad \boldsymbol{\mu}_{22} = \begin{bmatrix} 5 \\ 0 \end{bmatrix} \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1.2 & 0 \\ 0 & 1.3 \end{bmatrix}$$

$$\boldsymbol{\mu}_{31} = \begin{bmatrix} -2 \\ 4 \end{bmatrix} \qquad \boldsymbol{\mu}_{32} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \qquad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 2.0 & 0 \\ 0 & 1.5 \end{bmatrix}$$

The training set is shown in the scatter plot of figure 3.

The population densities are known and it is therefore possible to calculate the error rate for the classifier minimizing the Bayes decision rule with zero-one loss function. The results from using classical statistical methods, nonparametric regression based classifiers and MLP classifiers are shown in table 6. The statistical methods used are described in section 4 and the MLP classifiers in section 3.

All statistical analyses were done in the S-Plus language, using the `classif` library of B.D. Ripley and the `fda` library of T. Hastie and R. Tibshirani. The MLP analyses are done using the Aspirin/MIGRAINES package, Leighton (1992), and the `nnet` S-Plus library of B.D. Ripley. All S-Plus libraries are available from the statistics archives at Carnegie-Mellon University (`statlib@lib.stat.cmu.edu`).

The network trained with Aspirin/MIGRAINES (back-propagation with momentum) used a 0-1 sigmoid activation function for the hidden and output layers and a quadratic loss function. The validation set was used to choose the number of hidden nodes and to decide when to stop training. The networks trained with the `nnet` library had skip-layer connections from the input to the output layer. The results from the `nnet` library are denoted VM, because the optimization is done using the variable-metric or quasi-Newton method. Three combinations of output activation functions and loss functions were used; sigmoid units and quadratic loss (sig), sigmoid units and entropy loss (ent) and linear units with quadratic loss (lin). For the

Error Rates for the Three Class Synthetic Data Set			
Method	% wrong on training set	% wrong on test set	% sd.dev. on test set
Bayes decision rule	10.2	9.7*/13.8	1.4
Linear Discriminant Analysis (LDA)	19.3	21.0	1.7
Quadratic Discriminant Analysis (QDA)	11.3	13.7	1.4
1-Nearest Neighbour		16.7	1.5
3-Nearest Neighbour		14.0	1.4
Classification tree	5.7	13.3	1.4
Projection Pursuit Regression (SMART)	11.7	25.3	1.8
SMART+LDA	12.5	15.7	1.5
SMART+QDA	11.0	16.0	1.5
3-GLM logit	17.5	19.7	1.6
3-GAM logit	9.7	12.5	1.4
MARS(1)	10.8	13.8	1.4
MARS(2)	9.7	13.5	1.4
FDA/MARS (1)	11.3	15.0	1.5
FDA/MARS (2)	9.8	13.5	1.4
BRUTO	9.7	13.0	1.4
FDA/BRUTO	9.7	14.0	1.4
AM-MLP 2-6-3 (sig)	8.7	12.2	1.3
VM-MLP 2-6-3 (ent)	8.2	11.7	1.3
VM-MLP 2-3-3 (sig)	9.0	12.2	1.3
VM-MLP 2-3-3 (lin)	8.3	12.2	1.3

Table 6: Error rates and approximate standard deviations for the three class synthetic data set. The training set is of size 600. Model selection is performed using a validation set of size 600. The test set is also of size 600. The * denotes that the result was obtained using a test set of size 24 000. The approximate standard deviations quoted for the test set are based on assuming that the number of errors in the test set is binomially distributed. Pairwise comparisons have a 95 % confidence interval of half-length about 0.8 %, assuming that all errors made by the method with the lower error rate is made by the method with the higher error rate.

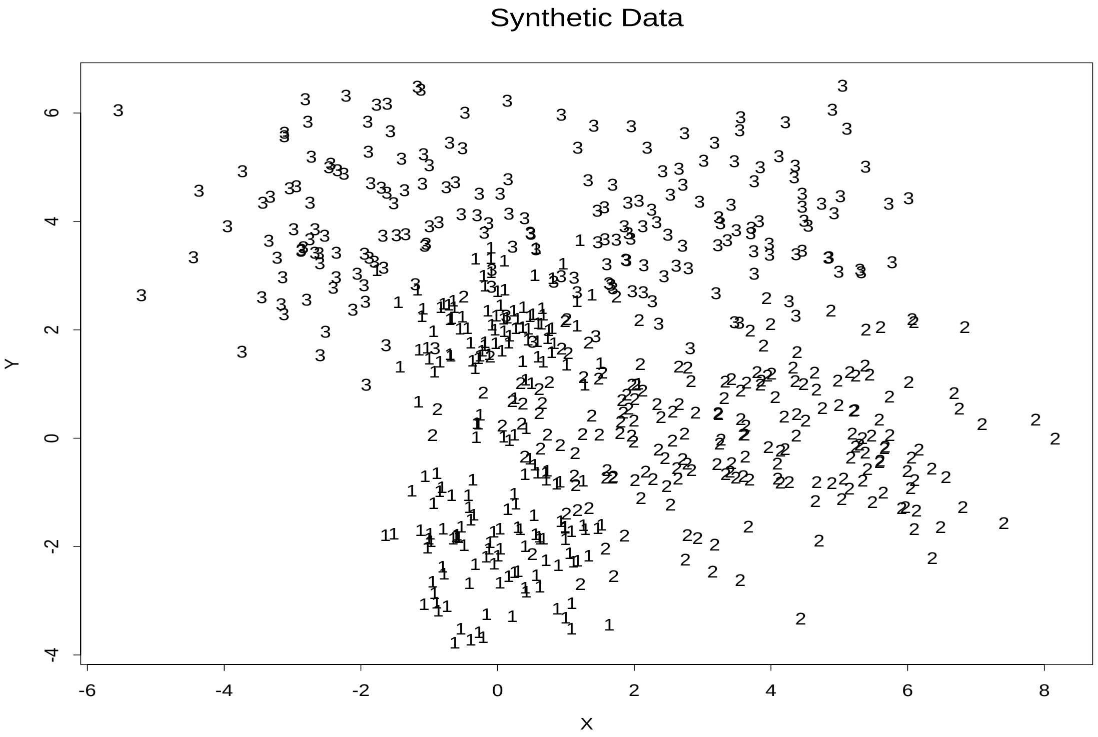

Figure 3: The three class synthetic training set.

hidden units sigmoid activation was used. The validation set was used to choose the number of hidden units using the error rate as the selection criterion.

There were small differences in the performance of the different classifiers. All the neural network methods performed well. From the plot of the training set, in figure 3, it was clear that linear methods would perform poorly.

Using McNemars test to test if the best MLP is significantly better than the results using the Bayes decision rule resulted in no significant difference between the methods (P-value of 0.14). McNemars test compares the number of mistakes made by method 1 and not by method 2 to the number of mistakes made by method 2 and not by method 1, see for example Fleiss (1981). These numbers are not seen in the table 6, where only the error rates are quoted.

Not all information is revealed using only one test set and quoting error rates. To visualize how the methods divide the total input space, a second data set was produced with 2500 points at a 50×50 lattice over the input space. This data set does not reflect the sampling distribution of the classes, but is an aid in visualizing the class boundaries for all the methods and in highlighting spurious effects. The resulting classification regions are shown in figures 4 and 5 for a selection of the classification methods.

The classification space produced by the Bayes decision rule can be regarded as the "true/best" classification space and the classification spaces produced by the other methods are compared with this space. Some of the plots (QDA, PPR, MARS(1), BRUTO) have class 2 regions on both side of the class 1 region. The QDA also has a class 1 region for x_2 above the plotting range. MARS (degree 1) has a strange spurious effect for large x_1 and x_2.

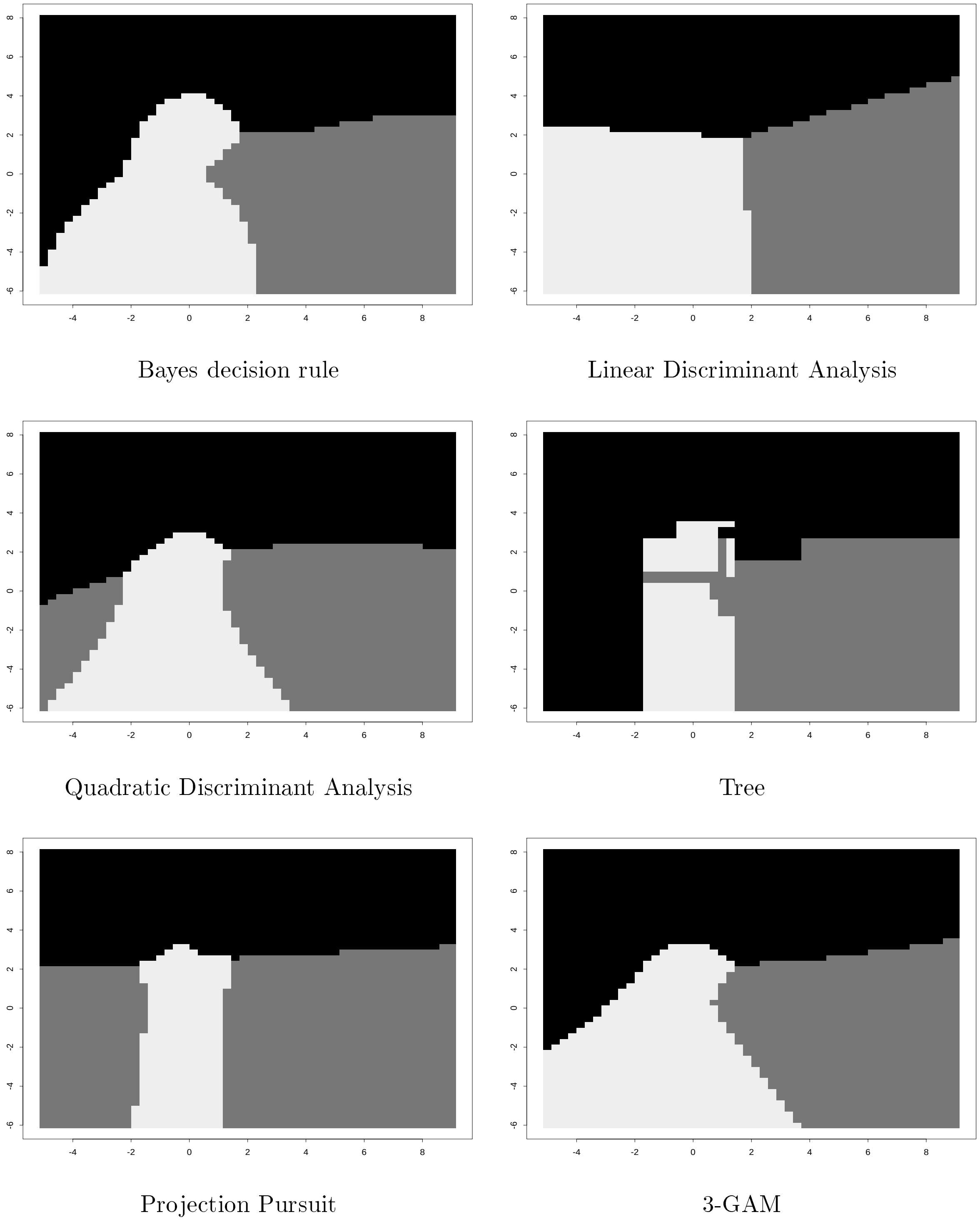

Figure 4: Classification regions for the three class synthetic data set. The light grey areas of the plots are class 1 regions, the dark grey areas are class 2 regions and the black areas are class 3 regions. The methods producing the classification regions are the Bayes decision rule, the linear discriminant analysis, the quadratic discriminant analysis, a classification tree, classification based on projection pursuit regression and classification based on 3 generalized additive models with logit link and Bernoulli responses.

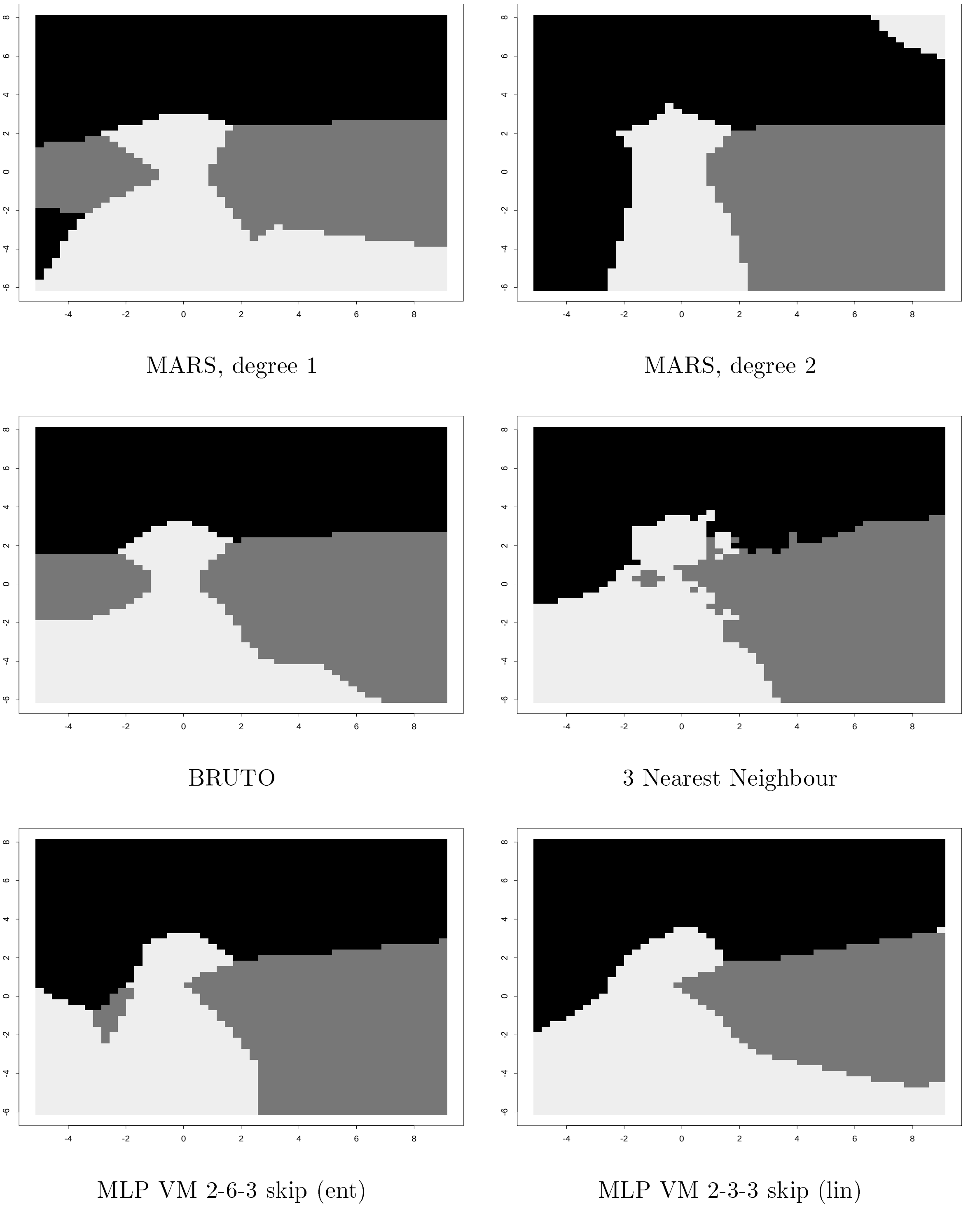

Figure 5: Classification regions for the three class synthetic data set. The light grey areas of the plots are class 1 regions, the dark grey areas are class 2 regions and the black areas are class 3 regions. The methods producing the classification regions are classification using Multivariate Adaptive Regression Splines (degree 1 and 2) and BRUTO, 3-nearest neighbour classification, a MLP with sigmoid output nodes and entropy loss and a MLP with linear output nodes and quadratic loss. Both MLP's have skip layer connections between the input and the output layer.

Based on both the plot of the classification space and the performance on the test set, the generalized additive model (3-GAM) and the MLP's (lin) and (sig) trained with VM are found to be the best methods in this situation.

7 Conclusions Based on Theoretical and Empirical Comparisons

Choosing a good model for a data set can be regarded as an art. McCullagh and Nelder (1989) give three principles to guide the data modeller. The first principle says that *all models are wrong, but some are more useful than others* (originally from Box (1979)). The second principle warns the data modeller to *not fall in love with one model to the exclusion of alternatives.* The third principle advocates the *use of diagnostic procedures to check how well the data fit the model.*

Another principle in data modelling, Friedman (1994), is that *no method dominates all others over all situations.* There is usually no uniformly best method. Each model has a set of situations where it works best.

The aim of this report has been to try to live by these principles when entering into the world of discrimination and classification. That has resulted in the presentation of methods from statistical regression, discrimination and classification, from pattern recognition and from static artificial neural networks. Methods from different fields are presented on a common platform.

From the theoretical presentation of the discrimination and classification methods in section 3 and 4 the conclusion is that nearly all the methods presented aim at estimating or approximating the posterior distribution of the classes. The methods differ in the possible forms of the decision boundaries they produce. Linear discriminant analysis, logistic discrimination, classification and regression trees and generalized linear models all produce linear (or at least piecewise linear) class boundaries. Quadratic discriminant analysis produces quadratic boundaries and kth nearest neighbour classification produces non-smooth and wiggly boundaries. The other methods presented all produce smooth and flexible class boundaries.

This presentation has included both parametric and nonparametric methods. Having decided the architecture and the activation functions, the MLP is a parametric model. The same is correct for LDA, QDA, logistic discrimination and CART. Projection pursuit regression, MARS and GAM are all able to use nonparametric spline functions and are therefore nonparametric in nature. From the empirical results found for the marine phytoplankton data set it may seem that the MLP's are able to fit the phytoplankton data better than flexible discriminant analysis. The problem of overfitting is also important and has caused problems for the MLP for the setting with the smallest data set of marine phytoplankton. When data is scarce, the simplest models are often the best, as for the two smallest data sets where LDA gave good results.

For most of the methods presented here, model selection is necessary. Some methods have selection methods incorporated in their algorithms; MARS (forward and backward steps), projection pursuit and CART (splitting and pruning). The other methods can make use of a separate validation set. Finding the best MLP architecture and the optimal training period is a tedious procedure. Search routines for finding a good network architecture (growing and pruning) exists, but does not make all model selection choices redundant (activation functions, loss functions, when to stop training)

In classification the aim is to predict the true class ω for an observation $\boldsymbol{X} = \boldsymbol{x}$. It is not necessary to know anything about the structure of the classifier, we can regard it as a *black box*. However, it is often desirable to interpret how the classifier works. That is, we want to visualize how the classifier separates the sample space into disjoint regions for the classes (discrimination). Then it is important to be able to produce plots of low dimensionality. This was possible for all the method for classification and discrimination used on the synthetic data set in section 6 because of the fact that the sample space was two-dimensional. For low dimensional sample spaces plots similar to those in section 6 will visualize how the sample space is separated into different classes. To use this approach with sample spaces of more than two dimensions, we need to plot projections of the sample space into two dimensions.

Some of the methods presented in this report have good interpretability. CART produces a binary tree where the class boundaries are found from the splits at the internal tree nodes. LDA ranks the discriminant functions on how well they separate the classes. Therefore plotting the first few discriminant functions will give good interpretability. The same is possible using flexible discriminant analysis. Projection pursuit regression models the discriminant functions as linear combinations of general functions of a set of projections (directions). The general functions can be plotted along these projections. MARS can be written as an ANOVA decomposition. This will give insight into the interactions modelled, but will not produce plots, see Friedman (1991). In a GAM the smoothers can be plotted along their univariate directions. Logistic discrimination produces linear class boundaries, but the discriminant functions are not ranked in order of separability. The two other methods presented in this report are also relatively difficult to interpret; kth nearest neighbour is constructed with raw observations in the training set and MLP consists of a hierarchy of functions of linear combinations.

Most of the methods presented in this report have difficulties in handling outliers and wrongly labelled observations. The methods that are based on loss functions (MLP, logistic discrimination, GAM, projection pursuit and MARS) have all the possibility of putting different weights on the observations in the training set and to add a penalty term to restrict the flexibility of the model fitted. In LDA and QDA it is possible to use robust estimates of the mean and covariance matrices. This is also the case for flexible discriminant analysis, using robust estimates in the transformed space. Data editing (which is mentioned in section 4.3) might be used to edit the training set. This is however dangerous, because valuable information can be deleted.

No accurate comparison with respect to the speed of the different methods is conducted. For the marine phytoplankton example it was clear that the flexible discriminant procedure is a much faster method than the multilayer perceptron.

Both the MLP and multi-response regression can be used in other problems than classification.

We will conclude this report by addressing some topics that could benefit from further study. Since the number of methods presented is large, not all methods are presented as thoroughly as they could have been. The problem of estimating class priors need more work. In the marine phytoplankton example both the MLP and the FDA methods used the class frequencies in the training set as an estimate for the class priors. It would be more interesting to modify the methods to include better estimates of the "true" class priors using expert opinion or alternatively to assume equal class priors. In this report only one data set from biology and one synthetic data set is presented. For the marine phytoplankton data set only two types of methods were used; MLP and FDA methods. For the synthetic data set a much larger variety of methods were tried. This reflects the fact that a real world data set needs much more work than a synthetic data set.

Acknowledgements

The marine phytoplankton data set is presented with the kind permission of Geir Johnsen and OCEANOR. I would like to thank Geir Johnsen and Oddgeir Samset for valuable help in understanding the marine phytoplankton data set. Also thanks to Robert Tibshirani and Trevor Hastie for help with the installation of the `fda` library. Thanks 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 the MLP Calculations on the Marine Phytoplankton Data Set

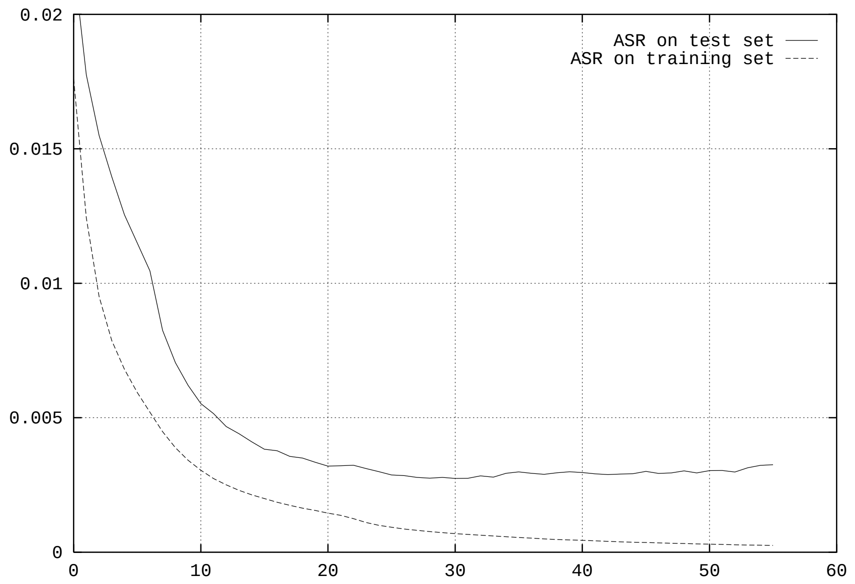

Figure 6: Plots of the ASR for the training and the test set for the the total marine phytoplankton data set with 3 inputs using a 3-15-31 MLP network. The network is trained for 14 000 epochs (one unit on the x-axis is approx. 250 epochs). The training set is of size 197 and the test set of size 20.

The Marine Phytoplankton Total Data Set with 3 Inputs 3-H-31 MLP networks after 5000 training epochs				
hidden nodes,H	ASR training set	% correct training set	ASR test set	% correct test set
4	0.013973	67.00	0.0196569	60.0
5	0.0106972	81.20	0.0174069	75.0
6	0.00830625	86.30	0.0143844	70.0
7	0.00698249	87.30	0.0136609	75.0
8	0.00658852	89.30	0.0119104	80.0
9	0.00459329	97.46	0.00815564	80.0
10	0.00364739	95.43	0.00808918	90.0
11	0.00347612	94.92	0.00865336	85.0
12	0.00247171	95.94	0.00719742	85.0
13	0.00192804	98.48	0.00658702	85.0
14	0.0019583	97.97	0.00757396	85.0
15	0.00155323	98.98	0.0033437	100.0
16	0.00145645	99.49	0.00713866	85.0
17	0.00117757	99.49	0.00503954	85.0
18	0.00116665	99.49	0.00599317	90.0
19	0.00122673	98.98	0.00560231	90.0
20	0.00123795	98.98	0.00454447	95.0

Table 7: Different MLP architectures for the total marine phytoplankton data set with 3 inputs. The MLP's are trained for 5000 epochs. The training set is of size 197 and the test set of size 20. The "ASR" is the average squared residual of the 0-1 encoding of the true class and the network output.

The Marine Phytoplankton Total Data Set with 3 Inputs																	
H	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
min(ASR)	3	20	19	18	18	20	20	19	19	20	20	20	19	19	19	20	20

Table 8: Number of epochs for which minimum ASR is obtained for the test set of the marine phytoplankton data set with three inputs using 3-H-31 MLP networks. The MLP's are trained using a training set of 197. The test set is of size 20.

The Marine Phytoplankton Total Data Set with 4 Inputs 4-H-31 MLP networks after 5000 training epochs				
hidden nodes,H	ASR training set	% correct training set	ASR test set	% correct test set
7	0.00557046	89.80	0.0107715	80.0
8	0.00385897	93.40	0.00915365	80.0
9	0.00302194	95.94	0.00936541	85.0
10	0.00220664	96.45	0.00962088	80.0
11	0.00146818	98.48	0.00631947	90.0
12	0.0018059	95.94	0.0018059	95.0
13	0.00100923	99.49	0.00100923	95.0
14	0.000884444	99.49	0.00688889	90.0
15	0.00064362	99.49	0.00561253	90.0
16	0.000666406	100.00	0.00407783	90.0
17	0.000526269	100.00	0.00579708	90.0
18	0.000625434	100.00	0.00533877	95.0
19	0.000703224	99.49	0.00457246	95.0
20	0.000529389	100.00	0.00416314	95.0

Table 9: Different MLP architectures for the total marine phytoplankton data set with 4 inputs. The MLP's are trained for 5000 epochs. The training set is of size 197 and the test set of size 20. The "ASR" is the average squared residual of the 0-1 encoding of the true class and the network output.

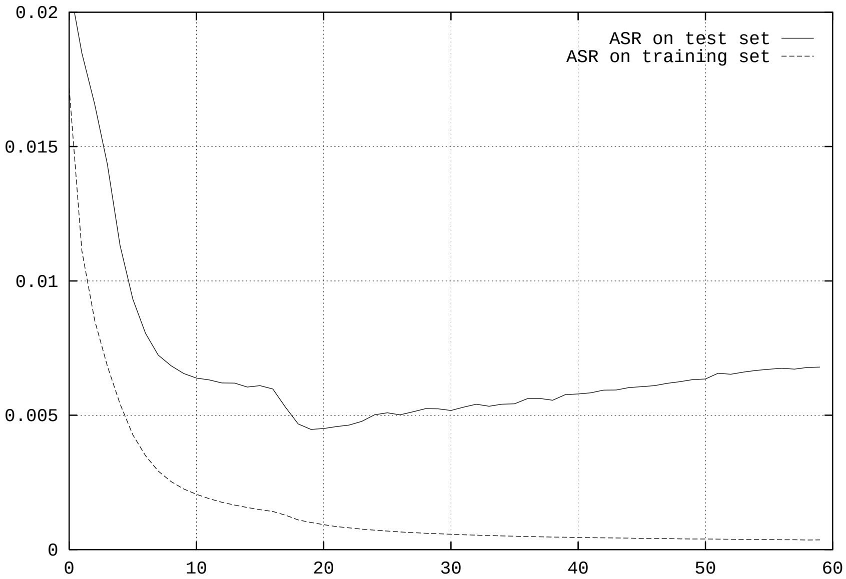

Figure 7: Plots of the ASR for the training and the test set for the the total marine phytoplankton data set with 4 inputs using a 4-13-31 MLP network. The network is trained for 14 000 epochs (one unit on the x-axis is approx. 250 epochs). The training set is of size 197 and the test set of size 20.

The Marine Phytoplankton High-Light Data Set 3-H-25 MLP networks after 10000 training epochs				
hidden nodes,H	ASR training set	% correct training set	ASR test set	% correct test set
1	0.0328126	35.60	0.0382856	10.00
2	0.0274138	45.80	0.0374561	20.00
3	0.0159951	79.70	0.00914207	60.00
4	0.0096484	93.22	0.0132224	60.00
5	0.00398445	94.92	0.00649993	60.00
6	0.00241822	98.31	0.00538386	80.00
7	0.00119924	98.31	0.00218851	90.00
8	0.000591243	100.00	0.0034311	70.00
9	0.000770938	100.00	0.00296252	80.00
10	0.000225444	100.00	0.00213419	70.00
11	0.00015195	100.00	0.00170846	90.00
12	0.000174877	100.00	0.00245396	80.00
13	0.000180007	100.00	0.00145888	90.00
14	0.000158823	100.00	0.00272376	70.00
15	0.00016068	100.00	0.00141257	80.00
16	0.00016133	100.00	0.0013408	90.00
17	0.000146159	100.00	0.00200802	80.00
18	0.000119327	100.00	0.00191746	90.00
19	0.000152008	100.00	0.00134928	90.00

Table 10: Different MLP architectures for the marine phytoplankton high-light data set. The MLP's are trained for 10000 epochs. The training set is of size 59 and the test set of size 10. The "ASR" is the average squared residual of the 0-1 encoding of the true class and the network output.

The Marine Phytoplankton Low-Light Data Set 3-H-28 MLP networks after 5000 training epochs				
hidden nodes,H	ASR training set	% correct training set	ASR test set	% correct test set
1	0.0222345	48.40	0.033422	42.85
2	0.0165458	64.50	0.0329253	47.62
3	0.0130103	77.40	0.0312767	66.67
4	0.00770813	88.70	0.0261994	80.95
5	0.00482594	93.55	0.0222907	90.48
6	0.00347192	95.16	0.016858	90.48
7	0.00186638	97.58	0.0223268	100.00
8	0.00186054	97.58	0.0106632	90.48
9	0.00123419	98.39	0.0138056	95.24
10	0.0012364	98.39	0.0164061	95.24
11	0.000872755	99.19	0.00912208	95.24
12	0.0010385	99.19	0.0108813	95.24
13	0.00086214	99.19	0.0138273	95.24
14	0.000886567	99.19	0.0181439	95.24
15	0.000958196	99.19	0.0179913	95.24
16	0.000840055	99.19	0.0197541	95.24
17	0.000849893	99.19	0.0109372	100.00
18	0.000938852	100.00	0.0145256	95.24
19	0.000906384	99.19	0.0189976	95.24

Table 11: Different MLP architectures for the marine phytoplankton low-light data set. The MLP's are trained for 5000 epochs. The training set is of size 124 and the test set of size 21. The "ASR" is the average squared residual of the 0-1 encoding of the true class and the network output.

References

Anderson, J. A. (1972), "Separate sample logistic discrimination", *Biometrika*, 59, 19–35.

Anderson, T. W. (1984), *An Introduction to Multivariate Statistical Analysis*, New York: Wiley, 2nd edition.

Barron, A. R. and Barron, R. L. (1988), "Statistical learning networks: A unifying view", in *Computer Science and Statistics: Proc. of the 20th Symposium of the Interface*, pages 192–203, Alexandria, Virginia: The American Statistical Association.

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

Box, G. E. P. (1979), "Robustness in the strategy of scientific model building", in Launer, R. L. and Wilkinson, G. N., editors, *Robustness in Statistics*, pages 201–236, New York: Academic Press.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*, Montery: Wadsworth and Brooks/Cole.

Breiman, L. and Ihaka, R. (1984), "Nonlinear discriminant analysis via ACE and scaling", Technical Report 40, Department of Statistics, University of California, Berkeley.

Bridle, J. S. (1989), "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition", in Fogleman-Soulie, F. and Hérault, J., editors, *Neuro-computing: Algorithms, Architectures and Applications*, New York: Springer.

Bridle, J. S. (1990), "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters", in Touretzky, D. S., editor, *Advances in Neural Information Processing Systems 2*, pages 211–217, San Mateo, California: Morgan Kaufmann.

Casella, G. and Berger, R. L. (1990), *Statistical Inference*, 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.

Cheng, B. and Titterington, D. M. (1994), "Neural networks: A review from a statistical perspective (with discussion)", *Statistical Science*, 9, 2–54.

Cover, T. M. and Hart, P. E. (1967), "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, IT-13, 21–27.

Cox, D. R. (1966), "Some procedures associated with the logistic qualitative response curve", in David, F. N., editor, *Research Papers in Statistics: Festscrift for J. Neyman*, New York: Wiley.

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

Dawid, A. P. (1976), "Properties of diagnostic data distributions", *Biometrics*, 32, 647–658.

Day, N. E. and Kerridge, D. F. (1967), "A general maximum likelihood discriminant", *Biometrics*, 23, 313–323.

Devijver, P. A. and Kittler, J. V. (1982), *Pattern Recognition: A Statistical Approach*, Englewood Cliffs, New Jersey: Prentice Hall.

Diaconis, P. and Shahshahani, M. (1984), "On non-linear functions of linear combinations", *SIAM Journal of Scientific and Statistical Computing*, 5, 175–191.

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

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

Efron, B. (1986), "How biased is the apparent error rate of a prediction rule", *Journal of the American Statistical Association*, 81, 461–470.

Fisher, R. A. (1936), "The use of multiple measurements in taxonomic problems", *Annals of Eugenics*, 7, 179–188.

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).

Fleiss, J. L. (1981), *Statistical Methods for Rates and Proportions*, New York: Wiley, 2nd edition.

Friedman, J. H. (1984), "SMART, User's Guide", Technical Report 1, Department of Statistics and Stanford Linear Accelerator Center, Stanford University.

Friedman, J. H. (1991), "Multivariate adaptive regression splines (with discussion)", *The Annals of Statistics*, 19, 1–141.

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.

Friedman, J. H. and Stuetzle, W. (1981), "Projection pursuit regression", *Journal of the American Statistical Association*, 76, 817–823.

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

Hastie, T. J. and Tibshirani, R. J. (1990), *Generalized Additive Models*, London: Chapman & Hall.

Hertz, J., Krogh, A., and Palmer, R. G. (1991), *Introduction to the Theory of Neural Computation*, Redwood City, California: Addison-Wesley.

Hjort, N. L. (1986), "Notes on the theory of statistical symbol recognition", Technical Report 778, Norwegian Computing Centre.

Huber, P. J. (1985), "Projection pursuit (with discussion)", *The Annals of Statistics*, 13, 435–525.

Hush, D. R. and Horne, B. G. (1993), "Progress in supervised neural networks", *IEEE Signal Processing Magazine*, January, 8–39.

Johnsen, G., Samset, O., Granskog, L., and Sakshaug, E. (1994), "*In vivo* absorption characteristics in 10 classes of bloom-forming phytoplankton: taxonomic characteristics and responses to photoadaptation by means of discriminant and HPLC analysis.", *Marine Ecology Progress Series*, 105, 149–157.

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

Leighton, R. R. (1992), *The Aspirin/MIGRAINES Neural Network Software, User's Manual, release V6.0*, Russell Leighton and the MITRE Corporation.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, London: Chapman and Hall, 2nd edition.

Minsky, M. and Papert, S. (1988), *Perceptrons: An Introduction to Computational Geometry*, Cambridge, Mass.: MIT Press, extended edition, first edition published in 1969.

Moody, J. E. (1992), "The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems", in Moody, J. E., Hanson, S. J., and Lippmann, R. P., editors, *Advances in Neural Information Processing Systems 4*, pages 847–854, San Mateo, California: Morgan Kaufmann.

Ripley, B. D. (1993a), "Neural networks and flexible regression and discrimination", in Mardia, K. V., editor, *Statistics and Images 2, Advances in Applied Statistics 2*, pages 39–57, Abingdon: Carfax.

Ripley, B. D. (1993b), "Statistical aspects of neural networks", in Barndorff-Nielsen, O. E., Jensen, J. L., and Kendall, W. S., editors, *Networks and Chaos - Statistical and Probabilistic Aspects*, pages 40–123, London: Chapman & Hall.

Ripley, B. D. (1994a), "Flexible non-linear approaches to 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 105–126, New York: Springer-Verlag.

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

Rosenblatt, F. (1958), "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review*, 65, 386–408, cited in Hush and Horne (1993).

Schiffmann, W., Joost, M., and Werner, R. (1994), "Optimization of the backpropagation algorithm for training multilayer perceptrons", Technical report, University of Koblenz, Institute of Physics, available from archive.cis.ohio-state.edu in /pub/neuroprose/schiff.bp_speedup.ps.Z.

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

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, London: Chapman & Hall.

Sjöberg, J. (1993), "Regularization issues in neural network models of dynamical systems", Licentiate Thesis No. 366, Department of Electrical Engineering, Linköping University.

Smith, C. A. B. (1947), "Some examples of discrimination", *Annals of Eugenics*, 13, 272–282.

Solla, S. A., Levin, E., and Fleisher, M. (1988), "Accelerated learning in layered neural networks", *Complex Systems*, 2, 625–639.

Stone, M. (1978), "Cross-validation: A review", *Mathematische Operationsforschung und Statistik, Series Statistics*, 9, 127–139.

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