Neural Networks and Related Methods for Classification
Author(s): B. D. Ripley
Source: *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 56, No. 3 (1994), pp. 409-456
Published by: Wiley for the Royal Statistical Society
Stable URL: http://www.jstor.org/stable/2346118
Accessed: 08-06-2018 17:11 UTC

J. R. Statist. Soc. B (1994)
56, *No.* 3, *pp.* 409–456

Neural Networks and Related Methods for Classification

By B. D. RIPLEY†

University of Oxford, UK

[*Read before* The Royal Statistical Society *at a meeting organized by the* Research Section *on Wednesday, October 20th, 1993,* Professor V. S. Isham *in the Chair*]

SUMMARY

Feed-forward neural networks are now widely used in classification problems, whereas non-linear methods of discrimination developed in the statistical field are much less widely known. A general framework for classification is set up within which methods from statistics, neural networks, pattern recognition and machine learning can be compared. Neural networks emerge as one of a class of flexible non-linear regression methods which can be used to classify via regression. Many interesting issues remain, including parameter estimation, the assessment of the classifiers and in algorithm development.

Keywords: CLASSIFICATION; GENERALIZATION; NEURAL NETWORKS; NON-LINEAR DISCRIMINANTS; PROJECTION PURSUIT REGRESSION; SEMIPARAMETRIC REGRESSION

1. INTRODUCTION

Neural networks have been developed rapidly since around 1985 and are now used widely. The subject area is not well defined, being distorted by funding considerations, but is dominated by two areas of study:

(a) feed-forward networks, also known as multilayer perceptrons, used for classification (and less often for regression and function approximation, e.g. in control; Miller *et al.*, 1990);
(b) symmetric recurrent networks, known as attractor neural networks or Hopfield nets, used as associative memories.

In both areas the biological connections were originally seminal but are now much weaker, and the emphasis is on solving application problems rather than on gaining insights into neurobiology. The second field is closer to biology and will not be discussed further in this paper. Amit (1989) is an excellent introduction. Another area of increasing activity is the use of recurrent neural networks to predict time series (e.g. Weigend and Gershenfeld (1994)).

There are several wide-ranging introductions to the field of neural networks, of which Hertz *et al.* (1991) is very suitable for a statistical audience. I have recently written an extensive review paper (Ripley, 1993) setting neural networks in a statistical context with examples. Weiss and Kulikowski (1991) contrast neural networks approaches with those of statistical pattern recognition and machine learning. One view that I have heard independently expressed several times is that the main impact

† *Address for correspondence*: Department of Statistics, University of Oxford, 1 South Parks Road, Oxford, OX1 3TG, UK.
E-mail: ripley@stats.ox.ac.uk

 0035–9246/94/56409

of neural networks has been to revitalize the field of pattern recognition, and the material in this paper could certainly be seen as part of that subject.

Classification and *discrimination* are used in a number of closely related senses. Consider the following real life examples:

(a) grading Danish bacon rashers;
(b) botanical field guide key to species identification;
(c) distinguishing male and female crabs of two species (Campbell and Mahon, 1974);
(d) recognizing symbols on hand-drawn maps (Hjort, 1986);
(e) predicting the occurrence of tsetse flies in Zimbabwe (Ripley, 1993).

In each we are given measurements (features) from a space $\mathscr{X}$ for each object and asked to assign a label from a set $\mathscr{G}$ of K classes. We shall usually think of $\mathscr{X}$ as a Euclidean space and of each dimension as a feature, but the framework includes discrete components and more complicated structures (such as whole images and point patterns of variable size). (Although we do not consider this point further until Section 8, readers should be aware that the major consideration in most pattern recognition problems is to select the 'right' features to make up $\mathscr{X}$.)

We may want a definite classification, or we may want some measure of belief in a possible classification, e.g. a probability distribution over $\mathscr{G}$. To allow for this additional information, we suppose that we are asked to supply a response from a space $\mathscr{Y}$. Typically this will be $[0, 1]^K$, but with ordered classes (as for bacon graded 1–4) we might return a latent continuous score. In any case, a *classifier* is a map $f: \mathscr{X} \to \mathscr{Y}$.

A classifier has to be learned (estimated) from a collection of p examples $(\mathbf{X}_i, Y_i)$ of features and the desired responses. The example response is usually a definite classification, but it need not be, especially when the responses are elicited from experts summarizing past experience.

There can be two rather distinct purposes in learning a classifier. The most obvious is to classify future observations, as in all five examples. However, for the crabs and tsetse flies we could be interested in the form of the classifier *per se*, i.e. in what combinations of the features discriminate between the classes. Only for predictive classification is a 'black box' representation of f useful. (However, it is sometimes possible to approximate the black box in a more interpretable way, e.g. as a set of decision rules. See Section 4.)

Problems of this general class (and often the very same examples) have been studied in the fields of pattern recognition, machine learning and neural networks as well as multivariate statistics. Although the precise methods of estimating the classifier f differ, the issues of model selection and assessment are common to all. In this paper we aim to lay out what is common and what is distinctive in the many approaches, to allow a synthesis of the best features of all. Some reviewers have commented that the paper contains surprisingly little about neural networks, but this represents a misunderstanding of the area; we do not expect a paper on regression to concentrate on linear least squares, and similarly the important developments in neural networks are not about the networks *per se* (Section 2) but how they are used. In such a fast developing field (emphasized by how recent the references are) I will surely have missed many things (or omitted them for lack of space) which the discussion will help to fill in.

2. FEED-FORWARD NEURAL NETWORKS

Feed-forward neural networks provide a flexible way to generalize linear regression functions. We start with the simplest but most common form with one hidden layer as shown in Fig. 1. The input units just provide a 'fan-out' and distribute the inputs to the 'hidden' units in the second layer. These units sum their inputs, add a constant (the 'bias') and take a fixed function ϕ_h of the result. The output units are of the same form, but with output function ϕ_o. Thus

$$y_k = \phi_o\left\{\alpha_k + \sum_j w_{jk}\phi_h\left(\alpha_j + \sum_i w_{ij}x_i\right)\right\}. \tag{1}$$

The 'activation function' ϕ_h of the hidden layer units is almost always taken to be the logistic function

$$l(z) = \exp z/(1+\exp z)$$

and the output units are linear, logistic or threshold units. (The threshold units have $\phi_o(x) = I(x>0)$.)

We can eliminate the biases α_i by introducing an input unit which is always at $+1$ and feeds every other unit. (This is the same idea as adding a constant column to the design matrix to include the intercept in a regression.) The function f is then parameterized by the set of weights w_{ij}, 1 for every link in the network (or 0 for links which are absent).

Such networks have a considerable history, including an original biological motivation, which is explained in Hertz *et al.* (1991). However, they can equally be seen as a way to parameterize a fairly general non-linear function from $\mathscr{X}$ to $\mathscr{Y}$. Such networks *are* rather general: Cybenko (1989), Funahashi (1989), Hornik *et al.* (1989) and later researchers have shown that neural networks with linear output units can approximate any continuous function f uniformly on compact sets, by increasing the size of the hidden layer. This implies approximation of measurable functions in measure and L_p. A short constructive proof (essentially that given by Jones (1990)) can be based on the denseness of trigonometric polynomials on a compact set in $\mathbf{R}^d$, the fact that a term of the form $\Pi_{i=1}^d \cos(\omega_i x_i + \psi_i)$ can be expressed as a sum of terms of the form $\cos(\alpha + \beta^T\mathbf{x})$, and the approximation of one-dimensional cosines by linear combinations of logistics, a one-dimensional version of the result.

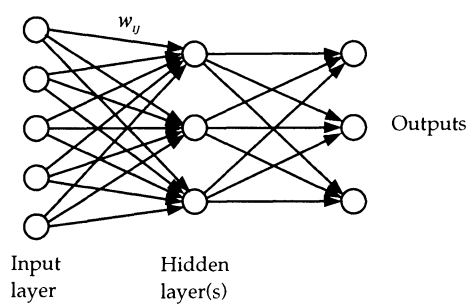

Fig. 1. Generic feed-forward neural network: normally all units in one layer are connected to all in the next layer, as shown

Since logistics can be approximated by step functions, the approximation results still apply with threshold units in the hidden layer.

Barron (1993a, b) and Jones (1992) show that (for sufficiently smooth f) the L_2-approximation is $O(N^{-1/2})$ where N is the number of hidden units. A heuristic reason why projection methods might work well with modest numbers of hidden units is that the first stage allows a projection onto a subspace of $\mathscr{X}$ of much lower dimensionality, within which the approximation can be performed.

Kůrková (1992) even has an approximation result for networks with two hidden layers and linear output units in which all the weights except those leading to the outputs are fixed, so the function is linearly parameterized.

These approximation results show that one hidden layer is sufficient, but it may be more parsimonious to use fewer hidden units connected in two or more hidden layers. Further, equation (1) does not include linear functions as a special case, although they can be approximated by scaling down all the weights w_{ij} and up all the weights w_{jo}, using the fact that the centre of the logistic function is linear. It may be preferable to include direct 'skip-layer' connections from the inputs to the outputs, obtaining

$$y_k = \phi_o\left\{\alpha_k + \sum_i w_{ik}x_i + \sum_j w_{jk}\phi_j\left(\alpha_j + \sum_i w_{ij}x_i\right)\right\}. \tag{2}$$

The approximation results apply to linear output units, but clearly they also apply to logistic output units if the component functions f_k are bounded away from 0 and 1. As we shall see, for classification it *may* be unimportant to approximate well at the extremes.

The approximation results are in general non-constructive, and in practice the weights have to be chosen to minimize some fitting criterion, e.g. least squares

$$E = \sum_p \| t^p - y^p \|^2$$

where t^p is the target and y^p the output for the pth example pattern. Statisticians may feel unhappy about using least squares to fit a function f_k with target values 0 and 1. Other measures have been proposed, included 'maximum likelihood' (Hinton (1989), Spackman (1992) and van Ooyen and Nienhuis (1992); in fact minus the logarithm of a conditional likelihood—see Section 6.1) or equivalently the Kullback–Leibler divergence (Solla *et al.*, 1988; Bridle, 1990), which amount to minimizing

$$E = \sum_p \sum_k \left\{ t_k^p \log\left(\frac{t_k^p}{y_k^p}\right) + (1 - t_k^p) \log\left(\frac{1 - t_k^p}{1 - y_k^p}\right)\right\}. \tag{3}$$

Another form is considered in Section 5.

The forms (1) and (2) and their multilayer extensions are attractive in that the derivatives of a fit criterion E with respect to the weights can be calculated recursively from output to input by using the chain rule, a procedure known as *back propagation*. This can be extended to calculating the Hessian (Bishop, 1991a, 1992; Buntine and Weigend, 1993). For the Hessian the forms are more complex but simplify somewhat for one hidden layer (including networks with skip-layer connections).

2.1. *Algorithms*

The classic algorithm of neural networks (also known as back propagation or the *generalized delta rule*) is to take fixed steps in the direction of steepest descent in minimizing the criterion E,

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}. \tag{4}$$

As the step control η is unchanged throughout the procedure, this has been seen as a form of stochastic approximation. Often 'momentum' is used, which can be seen as applying exponential smoothing to equation (4) leading to

$$\Delta w_{ij} = -(1-\alpha)\eta \frac{\partial E}{\partial w_{ij}} + \alpha(\Delta w_{ij})_{\text{old}} = -\eta' \frac{\partial E}{\partial w_{ij}} + \alpha(\Delta w_{ij})_{\text{old}}.$$

The fit criterion E will in all cases be a sum over terms E^p for each example pattern. Rather than to present all patterns, to compute $\partial E^p / \partial w_{ij}$ for each, sum and then to adjust the weights, we could adjust the weights after every pattern (presented in systematic order or randomly), a form even closer to stochastic approximation and in which momentum is desirable. (This is often known as the 'on-line' algorithm, in contrast with the 'batch' version.)

In my view too much attention has been spent on analogies with stochastic approximation (e.g. White (1989a, b)), which is not often used as a practical procedure in other problems. The problem is one of least squares or of minimizing a criterion close to least squares. As such it is amenable to general unconstrained optimization techniques (Fletcher, 1987; Gill *et al.*, 1981). For realistic numbers of weights (up to 1000) quasi-Newton methods work well. For larger problems the storage of the approximate Hessian can be too demanding, and conjugate gradient methods or the limited memory BFGS quasi-Newton method (Gill *et al.* (1981), p. 150) can be used. (Shanno (1990) reviews modern developments in limited memory optimization.) In my experience these methods work much better on hard fitting problems than those based on empirically chosen variants of steepest descent.

There are practical difficulties in minimizing E, since experience shows local minima to be very common. For fully connected networks there will be a number of equivalent sets of parameter values giving the same function f by symmetric interchanges of the hidden units, a mild lack of identifiability. In practice we find many minima with similar values of E corresponding to very different internal structures in the black box. (These have been discovered by the *multistart* algorithm, i.e. running the optimizer several times from randomly chosen starting sets of weights.) There has been some interest in global optimization procedures such as simulated annealing or a noisy variant of the stochastic approximation (Kushner, 1987; White, 1989a; Styblinski and Tang, 1990; Gelfand and Mitter, 1991) but these seem impracticable for routine use.

With the classic algorithm it is necessary to decide when to stop, and computational considerations may necessitate stopping before other algorithms reach a local minimum. It is common practice to stop when a high proportion (often 100%) of the training set is classified correctly, e.g. when 95% of the training set has output within distance 0.05 of the target. Such stopping rules can change quite dramatically

the success rate on a test set by avoiding overfitting and are a major potential for 'tuning' in comparative experiments. Another suggestion is to have a validation set (a test set used to tune the method but not to assess performance), and to stop when the error rate on the validation set starts to increase. (In my experience this often stops far too soon, before the minimum rate on the validation set is achieved on the path of weights taken in minimizing E.)

The computation of both $f(\mathbf{x})$ and $\partial E/\partial w_{ij}$ can be performed in almost any vector or matrix manipulation package, and macros have been written in many such packages to 'simulate' neural networks. Neural network simulator packages are often special purpose matrix manipulation packages, with graphical output facilities. For large problems more efficient code in a compiled language is needed. There are many freely available implementations of neural network procedures.

3. TWO-CLASS CLASSIFICATION PROBLEMS

For simplicity, let us assume for this section that there are just two classes labelled 0 and 1. This is quite a common case (predicting presence or absence) as well as providing some considerable theoretical simplifications.

The decision theoretic approach to classification is to assume a joint distribution for the features $\mathbf{X} \in \mathscr{X}$ and the class indicator $Y \in \{0, 1\}$, which we can specify by prior probabilities π_0 and π_1 for the class and class-specific distributions of features $p(\mathbf{x}|y)$. The Bayes rule is then to choose the class with the higher posterior probability $p(y|\mathbf{x})$. Rather than to look at the ratio $p(1|\mathbf{x})/p(0|\mathbf{x})$ let us first look at the difference

$$p(1|\mathbf{x}) - p(0|\mathbf{x}) = 2p(1|\mathbf{x}) - 1 = 2\{E(Y|\mathbf{X}=\mathbf{x}) - 0.5\}$$

which shows that the classifier can be found by thresholding $f(\mathbf{x}) = E(Y|\mathbf{X}=\mathbf{x})$ at $\frac{1}{2}$. Of course $\text{logit}\{f(\mathbf{x})\} = \log\{p(1|\mathbf{x})/p(0|\mathbf{x})\}$, recovering the logarithm of the natural ratio, and suggesting considering f on the logit scale. (It is easy to extend the discussion here to include general loss functions and 'doubt' and 'outlier' options, but for brevity these are omitted.)

In practice f will be an unknown function from $\mathscr{X}$ to $[0, 1]$ and so is an ideal candidate for approximation by a neural network with a single logistic output unit. This is the common practice, using least squares as the fitting criterion. For pure classification we are only interested in knowing whether $f(\mathbf{x}) > 0.5$, so an accurate approximation to f is only needed in the centre of its range. Note that our decision theory framework assumes that the densities are known, and uncertainties in parameters of the densities will affect the Bayes rule. For the time being we shall ignore these and take a 'plug-in' approach. (See also Section 6.4.)

If we are prepared to make assumptions of normality and equal covariance matrices for $P(\mathbf{x}|y)$, it is well known that

$$f(\mathbf{x}) = l(\alpha + \beta^{\mathrm{T}}\mathbf{x}), \tag{5}$$

i.e. logistic discrimination (as in Anderson (1982)). This can be fitted by maximum likelihood, which corresponds to fitting a neural network without a hidden layer using equation (3) (e.g. McLachlan (1992)). Thus the neural network model can be seen as a perturbation of logistic discrimination which allows for general mean functions $E(Y|\mathbf{X}=\mathbf{x})$. If the model is sufficiently complex to allow us to classify the whole training set correctly, we can then drive E to 0 by scaling up the weights to the output

units, so the existence of difficulties in parameter estimation for logistic regression (e.g. Lesaffre and Albert (1989)) reappear with a vengeance.

Other ways of fitting a more general function f can be derived from the recent statistical literature. Consider the classification of rock crabs (Campbell and Mahon, 1974). There are five measurements on the carapaces of the animals. Rather than to perform linear discrimination on the raw measurements, we might want to include ratios and so to perform the analysis on a logarithmic scale. Ideally, the classification procedure would itself choose the right scale. To do so we can consider a regression of the form

$$y = \alpha + \sum_{j=1}^{p} g_j(x_j) \tag{6}$$

for smooth but unknown functions g_j (Friedman and Silverman, 1989; Hastie and Tibshirani, 1990). If these are parameterized via splines with fixed knots, say, we have

$$y = \alpha + \sum_{j=1}^{p} \sum_{k=1}^{df_j} \beta_{jk} \phi_{jk}(x_j). \tag{7}$$

The smoothing procedure need not choose parameters by least squares, but if it does we have linear regression in an extended space $\mathscr{X}$ of features spanned by the functions $\phi_{jk}(x_j)$. We could include a logistic output unit and have

$$y = l\left\{\alpha + \sum_{j=1}^{p} g_j(x_j)\right\} \tag{8}$$

and fit by maximum likelihood, obtaining a generalized additive model (Hastie and Tibshirani, 1990). This idea applies to all the forms for f below.

These forms still do not allow interactions between the features in $\mathscr{X}$. Perhaps the simplest way to allow interactions is to allow linear combinations (projections):

$$y = \alpha + \sum_{j=1}^{r} g_j(\alpha_j + \beta_j^{\mathrm{T}} \mathbf{x}) \tag{9}$$

which is *projection pursuit regression* (PPR) (Friedman and Stuetzle, 1981). This is often written in the alternative form

$$y = \alpha + \sum_{j=1}^{r} \gamma_j \phi_j(\alpha_j + \beta_j^{\mathrm{T}} \mathbf{x})$$

for normalized (zero-mean, unit-variance) smooth functions ϕ_j. Equation (9) has the same approximation properties as the single-hidden-layer neural network (by the proof sketched in Section 2; see Diaconis and Shahshahani (1984) and Jones (1987)). Indeed, they are both based on projection directions, and in one dimension each can approximate the other. Zhao and Atkeson (1992) have an $O(N^{-1/2})$ approximation result for PPR over a larger class of functions than the analogous result (Barron, 1993a) for neural networks. Theoretical and empirical comparisons between PPR and neural networks are given by Hwang *et al.* (1992a, b, 1993).

Multivariate adaptive regression splines (MARS) (Friedman, 1991) allow for interactions more explicitly by

$$y = \alpha + \sum_{m=1}^{M} \beta_m \prod_{k=1}^{K_m} \phi_{km}(x_{\nu(k,m)}) \tag{10}$$

where $\phi_{km} = [x - t_{km}]_+$, $\phi_{k,m+1} = [t_{km} - x]_+$ and t_{km} is an observed value of $x_{\nu(k,m)}$. The *degree* is the largest K_m, the specific choices of functions being made to allow fast least squares fitting algorithms. Breiman (1991) has another class chosen with similar aims.

Another way to produce a flexible class of discrimination functions would be to parameterize $p(\mathbf{x}|y)$ as a mixture of a fixed set $\phi_j(\mathbf{x})$ of densities on $\mathscr{X}$, with weights p_{yj}. Then

$$p(\mathbf{x})\{p(1|\mathbf{x}) = p(0|\mathbf{x})\} = \sum_j \{\pi_1 p_{1j} - \pi_0 p_{0j}\}\phi_j(\mathbf{x}) \tag{11}$$

is a linear combination of the basis density functions and could be estimated. As $p(\mathbf{x})$ is unknown, this cannot be used directly to estimate the discriminant from the training sample, but it has the same sign as $f - \frac{1}{2}$ so can be used for classification. (Hall and Wand (1988) suggest estimating equation (11) by kernel smoothing.)

This also suggests parameterizing f or logit(f) as a linear combination of basis functions. For a one-dimensional $\mathbf{x}$ splines are a natural choice. For higher dimensions *radial basis functions* (RBFs) (Powell, 1987; Broomhead and Lowe, 1988; Moody and Darken, 1989; Poggio and Girosi, 1990; Musavi *et al.*, 1992) have been advocated. These are approximations of the form

$$y = \alpha + \sum_j \beta_j G(\|\mathbf{x} - \mathbf{x}_j\|) \tag{12}$$

for prespecified centres $\mathbf{x}_j$. Girosi and Poggio (1990) and Park and Sandberg (1991) showed approximation properties for this class, and Poggio and Girosi (1990) explored several issues in fitting. Examples of G proposed include the Gaussian $G(r) = \exp(-r^2/2)$ and the multiquadric $G(r) = \sqrt{(c^2 + r^2)}$. Alternatively we can use multi-dimensional splines (Wahba, 1990) to parameterize f.

Another approach to fitting $f(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$ would be to use nonparametric regression (e.g. Härdle (1990, 1991)), using kernel functions or nearest neighbours. However, we could use the analogous density estimation techniques to estimate $p(\mathbf{x}|y)$ and to compute the discriminant from $\pi_1 \hat{p}(\mathbf{x}|1)/\pi_0 \hat{p}(\mathbf{x}|0)$. This approach has been considered in monographs by Hand (1982) and Coomans and Broeckaert (1986). Estimating densities in high dimensions is notoriously difficult, and to use them for discrimination we are interested in regions where the densities of the two classes are comparable, usually in the tails of each. Methods of bandwidth selection, perhaps adaptively, should be tailored to the application. (Despite many publications on density estimation, there appears to be very little known for this problem.) Altogether it appears to be better to estimate the classifier directly. The approach of using a mixture distribution to represent a high dimensional density has perhaps been unfairly neglected, with exceptions (e.g. Roeder (1990)). The ideas have been seen as part of the neural network field: Specht (1990a, b) and Tråvén (1991).

4. SPACE PARTITION METHODS

Perhaps the most nonparametric method of all is that of k nearest neighbours, in which a majority vote is taken among the k nearest examples in $\mathscr{X}$ to the new unclassified example. This can be justified either via nearest neighbour density estimation, or as using the nearest neighbour nonparametric regression of Y on $\mathbf{X}$, but amounts to assuming local constancy of f and calculating a piecewise constant approximation. There are many practical issues, including the choice of a metric in $\mathscr{X}$ (which also applies to multidimensional kernel methods), the computation of fast searches for neighbours and the desirability of replacing the example patterns by a 'representative' set of points in $\mathscr{X}$. The last is the aim of *learning vector quantization* (Kohonen, 1990), which is considered to be a neural network method.

One idea that I have found rather fruitful (e.g. in the tsetse fly example in Ripley (1993)) is to use other non-linear discriminant methods (especially classification trees) to help to select the metric in $\mathscr{X}$, for instance omitting variables which do not appear in the tree, and weighting by their prominence if they do appear.

Classification trees also partition the space $\mathscr{X}$ into locally constant regions, often hypercubes parallel to the feature axes. There are many subtly different schemes for estimating trees, of which the best known to statisticians will be classification and regression trees (Breiman *et al.*, 1984), but perhaps the most influential is Quinlan's ID3 (Quinlan, 1979, 1983, 1986) from its seminal role in the machine learning community. Variants include those of Ciampi *et al.* (1987), Loh and Vanichsetakul (1988), Chou *et al.* (1989), Crawford (1989), Chan and Bao (1991), Chou (1991), Gelfand and Delp (1991), Gelfand *et al.* (1991), Buntine (1992), Clark and Pregibon (1992) and Fayyad and Irani (1992). The idea is to choose repeatedly a feature (or combination of features) and to split the space $\mathscr{X}$ on that value. The methods differ in allowing multiway splits or forcing binary splits and in how the best split is computed. More importantly, they differ in how to stop growing the tree and how to prune it to avoid overfitting.

Some tree construction methods allow linear or Boolean combinations of variables to define a split. Guo and Gelfand (1992) even allow a simple neural network at each node to select a non-linear combination of the feature variables there.

A tree can be expressed as a set of decision rules, one leading to each leaf. In his developments of ID3, now called C4.5, Quinlan (1990, 1993) simplifies these rules to increase generalization. Rules can also be extracted from neural networks (e.g. Gallant (1993), chapter 17). Other, direct, methods of inducing decision rules are considered in the field of machine learning. They have the considerable advantage of interpretability and so are particularly suitable for understanding the differences between the classes (see, for example, King *et al.* (1992)). They can also incorporate prior logical information much more easily than the methods described here.

Tree-based classifiers have been related to neural nets by Brent (1991) and Sethi (1990, 1991). They had essentially the same idea. Suppose that we have a binary tree with t splits and so $t+1$ leaves. The splits compute and threshold t functions of $\mathbf{x}$. Think of a network in which the first hidden layer of t nodes computes the split functions, and the second hidden layer has a node for each of the $t+1$ paths to a leaf which ANDs the (possibly inverted) outputs of the nodes on the path. The output layer then ORs leaves with the same class.

5. MORE THAN TWO CLASSES

In this section we shall specialize to a target set $\mathscr{Y} \subset \mathbf{R}^K$, and let $C \in \{1, \ldots, K\}$ denote the true class. For each $\mathbf{x} \in \mathscr{X}$ the classifier gives $y = f(\mathbf{x}) \in \mathscr{Y}$. We then have to decide how to turn y into a classification. For each class k the natural target for y_k is 1 on that class, 0 elsewhere, so we might want $f_k(\mathbf{x})$ to estimate $E(Y_k|\mathbf{X}=\mathbf{x}) = p(C=k|\mathbf{X}=\mathbf{x})$ where Y_k is the indicator of membership of class k. In that case the (approximate) Bayes rule is to choose the class k with the largest value of $f_k(\mathbf{x})$. As this depends only on the relative magnitudes of the components, it is invariant under any increasing transformation of f. Thus we can establish a general principle to choose the largest component of the classifier. Another idea (which is used in most of the neural network packages that I have studied) is to choose the exact classification (probability 1 on class k, 0 on the rest) which is nearest in the Euclidean space $\mathscr{Y}$ to the observed y. In some schemes, 'doubt' is reported if y is not closer than some prespecified tolerance (such as 0.2) to any exact classification. With the conventional targets, the nearest target *does* correspond to the largest component, but this scheme does not correspond to the Bayes rule with the doubt option, which declares doubt if no component exceeds a threshold.

If we regard the standard neural network procedure with two classes as non-linear logistic regression, the natural extension to more than two classes is a log-linear model for the posterior probabilities, which gives rise to

$$p(k|\mathbf{x}) = \frac{\exp f_k(\mathbf{x};\, \hat{\theta})}{\sum_j \exp f_j(\mathbf{x};\, \hat{\theta})} \tag{13}$$

and (conditional) negative log-likelihood

$$E = \sum_k t_k \log\left\{\frac{t_k}{p(k|\mathbf{x})}\right\}. \tag{14}$$

Here f can be any flexible family of multivariate functions such as neural networks with *linear* output units. (As equation (14) is not least squares, this procedure is less suitable for families such as MARS which are designed for fast least squares fitting.) In the neural network literature the log-linear approach is often called *softmax* after Bridle (1989, 1990).

Note that this differs from the commonest approach using *logistic* output units, which can be seen as fitting separate logistic models for each class *versus* the rest, without imposing consistency conditions on the posterior probabilities.

The component functions f_k can be estimated as individual regressions, or the function f by multivariate regression. The difference will only appear in the more general forms of regression of Section 3, in which we have the choice of using the same basis functions for each regression or not. Feed-forward neural networks are naturally expressed with a common set of intermediate values in the hidden layer, but the weights could be chosen with a pattern of 0s so that each output has its own set of hidden units. For PPR we could have any of

$$y_k = \alpha_k + \sum_{j=1}^{r} \gamma_{jk} \phi_j(\alpha_j + \beta_j^{\mathrm{T}} \mathbf{x}),$$

$$y_k = \alpha_k + \sum_{j=1}^{r} \gamma_{jk} \phi_{jk}(\alpha_j + \beta_j^{\mathrm{T}} \mathbf{x}),$$

$$y_k = \alpha_k + \sum_{j=1}^{r} \gamma_{jk} \phi_{jk}(\alpha_{jk} + \beta_{jk}^{\mathrm{T}} \mathbf{x}),$$

the last corresponding to completely separate regressions. The S-PLUS implementation uses the first, which is the most parsimonious and hence the most useful. The second allows the idea of using projections to pick an r-dimensional flat (translated subspace) in $\mathscr{X}$, and approximating within that flat.

For two classes, linear regression gives the sample linear discriminant if the classes are properly represented in the training sample. We can think of the linear regression as the best (least squares) linear approximation to the posterior probabilities. As a principle of classifier design this has been used (Duda and Hart, 1973; Devijver and Kittler, 1982; Fukunaga, 1990) under the name of *minimum (mean) square error* classifiers. Unlike the linear discriminant, that procedure classifies by the nearest target or equivalently the largest component. The difference is small with two classes; the linear combination is the same, but the threshold in linear discrimination may differ from 0.5 (although usually negligibly; Ripley and Hjort (1994), section 3.3).

In unpublished work, Breiman and Ihaka (1984) showed how to use regression to find the linear classifier for more than two classes. Their idea is to replace the targets y by a $(K-1)$-dimensional score $\theta(y)$ and to fit by least squares in the space of θ-scores. (Think of putting the three targets for a three-class problem around the circumference of a circle in $\mathbf{R}^2$.) Then, if we optimize over the choice of scores as well as the parameters in the regressions, the sample linear discriminant is recovered up to computable scale factors. What is not made clear in their work is that this process is equivalent to performing a linear discriminant analysis in the space of fitted values from the regressions of the class indicators on $\mathbf{x}$ (Ripley and Hjort, 1994).

The motivation of the Breiman–Ihaka approach was to use semiparametric rather than linear regression; we can set up a flexible semiparametric classifier, by applying the linear classifier to the space of fitted values, or by optimizing over the scores while fitting the function $f(\mathbf{x})$ by least squares. (If the regression is linear in y, such as a neural network with linear output units, the optimization over scores can be done without refitting the regression.)

This approach allows us to approximate optimally the discriminant in $J < K-1$ dimensions by using the first J discriminants. This can be used to present the data on the first few generalized discriminant variables, and that can be revealing (see Section 6.3 and Fig. 10 later). However, there can be difficulties with non-linear methods fitting the training set extremely well, and hence the information in the residuals reflects the stopping rule rather than a genuine pattern of lack of fit. Even when the lack of fit is genuine, it can be dominated by outliers, and a robust linear discriminant analysis applied to the fitted values can be helpful (Ripley and Hjort, 1994).

These ideas have been independently taken up by Hastie, Buja and Tibshirani (1992)

and Hastie, Tibshirani and Buja (1992), who extend them to penalized regression methods such as additive models with smoothing spline smoothers.

6. ISSUES IN FUNCTION FITTING

The following issues apply to all the methods that compute and maximize over discriminant functions.

6.1. *Parameter Estimation*

Maximum likelihood estimation of parameters in classifier functions $f(\mathbf{x}; \theta)$ is mentioned frequently in the literature. We need to consider the likelihood a little more carefully. That usually used for generalized linear models corresponds to fixing the samples from $\mathscr{X}$ and only considering the uncertainty in the responses. For discrimination it is more natural to consider either that $(\mathbf{X}_i, Y_i)$ are random samples from $p(\mathbf{x}, y)$, *mixture sampling*, or that the numbers in each class are fixed, so that *separate samples* are taken from each $p(\mathbf{x}|y)$. It is known (e.g. McLachlan (1992)) that these sampling schemes reduce to the same maximum likelihood problem *provided that*

$$p(\mathbf{x}, y) = p_\theta(y|\mathbf{x})\, p(\mathbf{x}),$$

i.e. that the marginal distribution of the $\mathbf{X}_i$ is independent of the parameters. This is *not* the case for the normal model for linear discrimination. The maximum likelihood method (3) of fitting the logistic (5) is based on the conditional likelihood of Y given $\mathbf{X}$ and so may be inefficient relative to estimating the parameters in the joint distribution by using the full likelihood. McLachlan (1992), section 8.5, summarizes work on assessing this inefficiency. This work is asymptotic, and for classification the effect of parameter estimation is asymptotically negligible, so the results are about fine distinctions. In practice we are a very long way from asymptotics, and the performances can show important differences.

The Fisher (1936) sample linear discriminant is fully efficient for the normal model and may be calculated via linear regression. However, the logistic form has wider validity and may be more efficient under other models for $p(\mathbf{x}, y)$. Thus the regression approach gives a fully efficient estimator of the classifier under the equal covariance matrix normal model, but there are no claims that this extends to the non-linear versions.

6.2. *Degree of Smoothness and Complexity*

The aim is to achieve the best classifier on the whole set $\mathscr{X}$, not just on the training set, and with a flexible fitting procedure for f there is the ever-present danger of overfitting. Stopping rules, as we have seen, are an *ad hoc* way to avoid this. In Section 7 we shall consider the general problem of doing well over all of $\mathscr{X}$; here we look at ways to choose a smooth f.

One way to ensure that f is smooth is to restrict the class of estimates, for example by using a limited number of spline knots or RBFs. Another way is *regularization* (Poggio and Girosi, 1990; Bishop, 1991b, 1993) in which the fit criterion is altered to

$$E + \lambda C(f)$$

for example, with a penalty C on the second derivatives of f such as

$$\frac{1}{p}\sum_{\mathrm{i,o}}\frac{\partial^2 y_{\mathrm{o}}}{\partial x_i^2}.$$

Such norms are used in the derivation of splines and RBF networks. *Weight decay*, specific to neural networks, uses as penalty the sum of squares of the weights w_{ij}. (This only makes sense if the inputs are rescaled to range about [0, 1] to be comparable with the outputs of internal units.)

These ideas can be seen as generalizations of ridge regression and shrinkage methods, and like them derived from a Bayesian perspective, by taking a prior proportional to $\exp(-\lambda C)$. In the neural network field this has been taken up by MacKay (1992) following Gull (1989). However, they appear to regard the choice of the parameter λ as a model choice, and so select a maximum of its posterior, and plug the estimate in. A wider ranging discussion is given by Buntine and Weigend (1991) who advocate a hyperprior on λ and integration rather than maximization. For weight decay, the prior corresponds to independent Gaussian-distributed weights of mean 0 and variance $\sigma_{\mathrm{w}}^2 = 2/\lambda$. Then MacKay fixes σ_{w}^2 whereas Buntine and Weigend integrate over a non-informative hyperprior. At least in this case, replacing the average over the posterior for σ_{w}^2 by the integrand evaluated at its maximum seems to be a rather poor approximation. (See the reply to the discussion.)

An issue related to the degree of smoothing is how many terms to allow in the model, e.g. how many hidden units in a neural network or how many functions g_j in PPR (9). These do not necessarily control the smoothness (which in PPR is primarily controlled by the smoothness of the functions) as much as the dimensionality of the approximation.

Several techniques have been proposed to select the model complexity. In PPR the number of terms in equation (9) is controlled by fitting additional terms one at a time, *keeping all other parameters fixed*, then using simultaneous fitting and backward selection. The method of *cascade correlation* (Fahlman and Lebiere, 1990) is similar in adding hidden units one at a time but omits the simultaneous fitting (although this could easily be done). In fact, the hidden units are each added in a new layer, connected to all earlier units, whereas in PPR a new term is fitted to residuals from the existing fit. Other constructive techniques are described in chapter 10 of Gallant (1993).

Another approach is to penalize model complexity as in Mallows's C_p and Akaike's AIC. One difficulty with such criteria for neural network models is to quantify the appropriate complexity, which is not simply the number of parameters (Moody, 1992; Murata *et al.*, 1991; Amari and Murata, 1993; Barron, 1993b).

Many methods ultimately resolve their free parameters for model smoothness and complexity by some form of cross-validation, either by using a separate validation set or by dividing (and rotating) the training set. The objective used in cross-validation is often least squares, even though the actual goal is classification for which it is more important to know the discriminant well near its threshold(s). My impression is that the use of cross-validation ideas in these non-linear and highly parameterized problems is not fully understood; in practice the variability of cross-validation estimates causes difficulties.

6.3. *Robust Fitting*

It is common practice to find that a small number of the example patterns are wrongly classified, and thus scored as 1 rather than 0 or vice versa. For example, in the iris data (Anderson, 1935; Fisher, 1936) the non-linear discriminant analysis given in Ripley (1993) throws doubt on the classification of three of the 150 examples.

Whereas the maximum error that can be made is 1, the fitting methods will usually fit the training set very accurately, and so the error will be many times the residual standard error. From the least squares point of view it is essential to use a robust criterion if overfitting is to be avoided. From the point of view of logistic regression we have the possibility of 'bit flip' errors, for which resistant methods have been proposed (Pregibon, 1982; Copas, 1988; Carroll and Pederson, 1993).

Most of the discussion of robustness has been for problems with small numbers of parameters compared with the number of data points, using asymptotic expansions, and it would be very helpful to have proven methods for semiparametric models. The issues are rather different, as a misclassification can have only a local effect on the fitted surface, and outliers in the $\mathscr{X}$-space can have only very limited leverage. The major impact of errors in the training set is on the choice of model complexity and on the assessment of the true performance level.

6.4. *Predictive Bayesian Approach*

In the Bayesian approach parameter estimation is not an issue in itself; we need to compute $p(y|\mathbf{x}, \mathscr{T})$ for the training set $\mathscr{T}=\{(\mathbf{X}_i, Y_i), i=1, \ldots, p\}$ and hence to compute the classifier. Here $\mathbf{x}$ is an arbitrary future observed pattern. Suppose that we have parameters θ in the model for $p(\mathbf{x}, y; \theta)$. Then

$$p(\theta|\mathscr{T}) \propto p(\theta)\, p(\mathscr{T}|\theta) = p(\theta) \prod_{i=1}^{p} p(\mathbf{x};\, \theta) f_{y_i}(\mathbf{x};\, \theta)$$

from which we can deduce that

$$p(y|\mathbf{x}, \mathscr{T}) = \int p(\theta,\, y|\mathbf{x}, \mathscr{T})\, \mathrm{d}\theta = \int p(y|\mathbf{x};\, \theta)\, p(\theta|\mathscr{T})\, \mathrm{d}\theta.$$

This approach is sometimes called the predictive approach (Aitchison and Dunsmore, 1975). It may appear computationally prohibitively expensive for neural network models for $p(y|\mathbf{x}; \theta)$, but Markov chain Monte Carlo methods have recently been used by Neal (1992, 1993) to compute the predictive distribution in a small (regression) example. As in Section 6.2 we may also use integration over hyperparameters in $p(\theta)$, which will affect the formula for $p(\theta|\mathscr{T})$ and may need a further level of Monte Carlo integration.

Even when finding the full predictive distribution is computationally prohibitive, we can use the theory to suggest useful hints. For example, when fitting neural networks we often encounter local minima. The theory suggests that the correct way to proceed is to average the predictive probabilities given by the solutions, weighted by the probability of the peak around $p(\hat{\theta}|\mathscr{T})$, and that can be approximated by the frequency with which that solution occurs in randomly started optimization runs. (A version of this idea for trees is implemented by Buntine (1992).)

7. ASSESSING GENERALIZATION

The real test of how a classifier f performs is its behaviour over the whole of the feature space $\mathscr{X}$, typically measured by the average misclassification rate

$$P[f(\mathbf{X}) \neq \arg\max_k \{P(C=k|\mathbf{X}, \mathscr{T})\}]$$

averaged over the unconditional distribution $p(\mathbf{x}) = \Sigma \pi_k p(\mathbf{x}|k)$, although we may also wish to consider some form of worst case behaviour over $\mathbf{x} \in \mathscr{X}$. Following the terminology of psychology this idea is referred to as *generalization ability*.

Of course, asymptotically in the number p of example patterns, we would expect the parameter estimation to be irrelevant provided that f is estimated in a finite dimensional space, or in a space growing sufficiently slowly with p, and theorems to this effect are given by White (1989b, 1992), although, as he states that at least 100 times as many example patterns as parameters are needed for these results to be relevant, they seldom will be. Geman *et al.* (1992) discuss this in the general context of nonparametric regression, but point out (pages 33 and 44) the irrelevance of asymptotic theory to most practical pattern recognition problems, in which we have to establish a compromise between bias and variance in estimating the true classifier. Stone (1982) has a general treatment of the rates of approximation which can be achieved in nonparametric regression. Donoho and Johnstone (1989) consider a rather restricted class of two-dimensional functions and show that projection-based and local (kernel-based) approximation methods have domains in which each is much better suited than the other, so there will be no universally best methods. We might hope that asymptotic theory would allow us to approximate $P\{C=k|\mathbf{x}, (\mathbf{x}_i, y_i), i=1, \ldots, p\}$ by averaging $P(C=k|\mathbf{x}, \theta)$ locally about optimal fitted θ (Buntine and Weigend, 1991), but this seems a forlorn hope given the difficulty of maximization and many good local minima found in practice. For sufficiently large training sets we can apply asymptotic theory as usual. (Buntine and Weigend (1991) say that 'the sample size should be at least some factor of the number of weights in the network'—I presume that they meant multiple rather than factor.) Other forms of approximation are those of Rissanen (1987) and Wallace and Freeman (1987).

In the remainder of this section we sketch a method to tackle the scaling of the generalization ability of suboptimal procedures with the size of the training set, which can be used to provide sufficient bounds on the size of the training set required. Barron has developed complexity measures (surveyed in Barron (1993b)) based on this method.

Consider a two-class classification problem, and let f be the correct classification function to be learned from p examples. For any function $\phi: \mathscr{X} \to \{0, 1\}$ define

$$g(\phi) = P\{\phi(\mathbf{X}) \neq f(\mathbf{X})\},$$

$$g_p(\phi) = \#\{\phi(\mathbf{x}_i) \neq f(\mathbf{x}_i), i=1, \ldots, p\}/p$$

where $\mathbf{X}$ is a random element of $\mathscr{X}$ from a prespecified distribution. Thus $g(\phi)$ is the misclassification rate for classifier ϕ, and $g_p(\phi)$ is the apparent misclassification rate on the training set. We know that $g_p(\hat{f})$ will be negatively biased for $g(\hat{f})$, but we can use a worst case bound on their difference.

Consider a training set randomly chosen from the same distribution as the test cases. We have (Blumer *et al.*, 1989; Abu-Mostafa, 1989; Vapnik, 1982)

$$\delta = P\{\max_{\phi} |g_p(\phi) - g(\phi)| > \epsilon\} \leqslant 4\{(2p)^{d_{VC}} + 1\}\exp(-\epsilon^2 p/8) \quad (15)$$

where $d < \infty$ is the Vapnik–Chervonenkis (VC) dimension of the space of exactly representable binary functions of the classifications of p examples. (There are other similar bounds, some linear in ϵ.) Thus

$$|g(\hat{f}) - g_p(\hat{f})| < \epsilon$$

with high probability if the right-hand side of inequality (15) is small, which it will be for large p. (This can be translated to a sufficient bound on p of order $(8d/\epsilon)\log(13/\epsilon)$.) In that case, if we can do well on the training set, we can also have confidence that usually we shall do well on future examples. Note that the bound is independent of the distribution over $\mathscr{X}$. There are also necessary bounds on p of order $d/32\epsilon$ when $g_p(\hat{f}) = 0$, i.e. that the training examples are fitted exactly.

Baum and Haussler (1989) applied these ideas to feed-forward networks with M *threshold* nodes and W weights. They showed that

$$d \leqslant 2W\log_2(eM)$$

and for a single-hidden-layer network with N inputs and H nodes in the hidden layer (M also includes output nodes) that

$$d \geqslant 2\lfloor H/2 \rfloor N \approx W$$

for large H. Thus the size of the training set needs to be of order W/ϵ to have a high probability of success rate at most ϵ worse on the test set than on the training set.

This is an example of Valiant's (1984) concept of *PAC learning*, where PAC stands for 'probably almost correct'. The idea is to seek a procedure which with arbitrarily high probability $1 - \delta$ has misclassification rate less than ϵ and takes polynomial time in the size of the problem and $1/\delta$, $1/\epsilon$. We have the stark result (Judd (1990), corollary 22)

'Networks cannot generalize'

(Judd has a technical condition on *loading* which some reviewers see as restrictive), the apparent contradiction being resolved by the restriction on the class of classifiers that the Baum–Haussler network can learn, and the absence of an explicit algorithm for training such networks. At present positive applications of the PAC concept are confined to functions f which can be represented by the network and to networks of threshold units. An empirical study (Cohn and Tesauro, 1992) shows error rates that are far below the upper bounds even in such networks. Baum (1990) gives theoretical support that the bound over all distributions of examples is too conservative. For neural networks with logistic units the VC dimension has only very recently been shown to be finite (Macintyre and Sontag, 1993) and no bounds are known.

8. EXAMPLES

Our three examples illustrate some of the intricacies of non-linear classification. They illustrate two extremes of the range of example problems. Few of the computations took longer than 2 min on a Sun SparcStation IPC, the exceptions being some of the cross-validation runs and the larger MARS and neural network experiments on the full sonar data which took up to 20 min each.

Performance is assessed on separate test sets. It would be possible to assess performance by more sophisticated techniques from the training set, e.g. by using estimated posterior probabilities (Fukunaga, 1990) or bootstrapping (Weiss and Kulikowski, 1991). However, the comparisons would be distorted by biases in these techniques which appear to differ considerably between types of classifiers.

8.1. *Synthetic Data*

The test set is illustrated in Fig. 2. Each population is an equal mixture of two two-dimensional normally distributed populations, and the two populations are equally likely. The boundary of the Bayes rule is shown, calculated from the known population densities.

The decision boundaries found for a training set of size 250 are shown in Figs 3–6. The error rates were estimated by using a further sample of size 1000 and are given in Table 1. Even with this size of test set the quoted rates have a standard error of about 1%. This example is rather noisy, and this favours the smoothing methods over the space partition methods. Notice the improvement that editing (Devijver and Kittler, 1982) gives for the nearest neighbour rule; unfortunately editing is only particularly effective in small numbers of dimensions with large training sets.

Fig. 4 shows the dramatic effect of weight decay (set at $\lambda = 5 \times 10^{-3}$) in smoothing the discriminant function, and this is reflected in the considerably better performance in Table 1. Note that for this training set PPR appears to fit rather better than neural networks, and that the non-differentiability of the function fitted by MARS shows clearly in Fig. 6. As far as possible the parameters (such as the number of learning

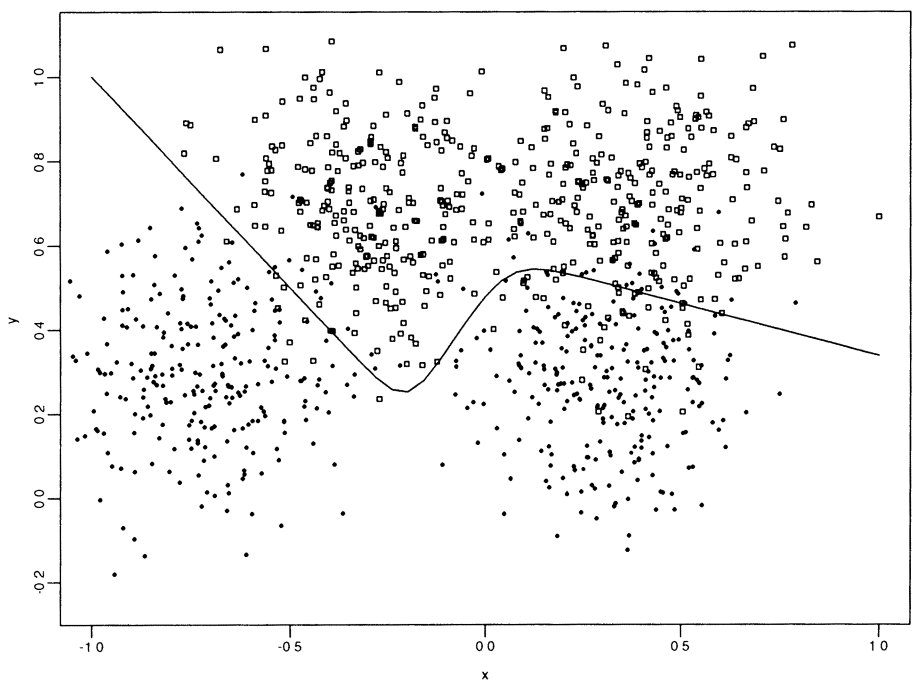

Fig. 2. Two-dimensional synthetic data set with the boundary of the Bayes rule (this is the test set of size 1000): the two classes are shown by ● and □

TABLE 1
Estimated error rates for the synthetic data set based on a test set of size 1000†

Method	*Error rate (%)*	*Method*	*Error rate (%)*
Bayes rule	8.0	Neural network with 3 hidden units	9.4
Linear discriminant	10.8	Neural network with 3 hidden units without weight decay	11.1
Logistic discriminant	11.4		
Quadratic discriminant	10.2	Neural network with 6 hidden units	9.5
1 nearest neighbour	15.0	PPR ($r=2$)	8.6
3 nearest neighbour	13.4	MARS (degree 1)	9.3
5 nearest neighbour	13.0	MARS (degree 2)	9.6
5 nearest neighbour after multiedit	8.3	Classification tree	10.1
		Learning vector quantization (12 representatives)	9.5

†Standard errors are around 1%, and pairwise comparisons have a 95% confidence interval of half-length about 0.7%.

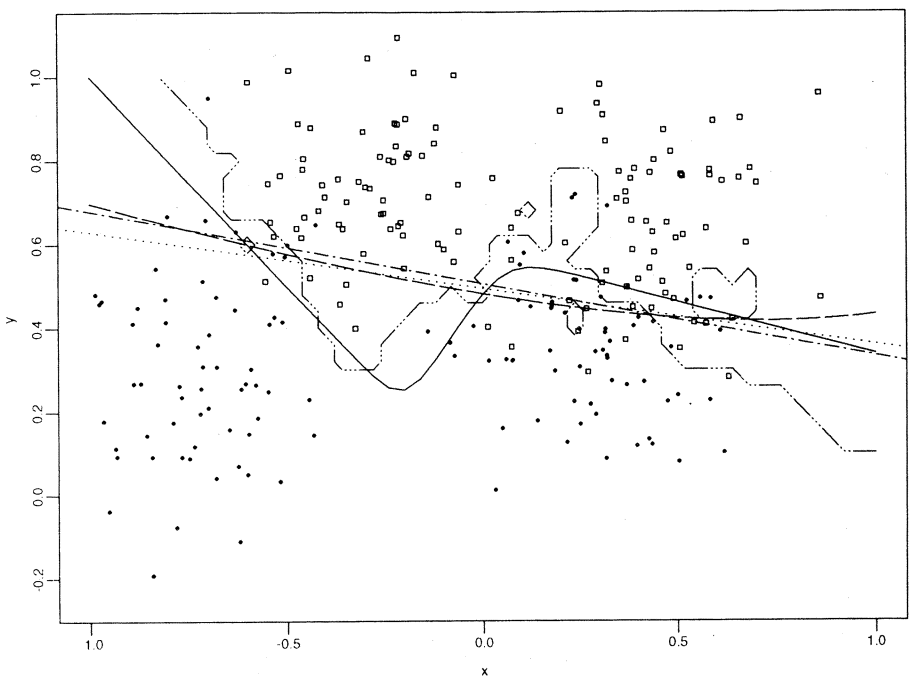

Fig. 3. Training set with the Bayes rule (———), linear (· · · · ·), logistic (– · –) and quadratic (— —) discriminants and the five nearest neighbour rule (— · · ·)

vector quantization representatives) were chosen automatically or the defaults were used; in some cases I did some preliminary testing on a further validation set of size 250.

8.2. *Sonar Data*

Gorman and Sejnowski (1988) published an early application of neural networks, which is one of White's (1989b) examples of

> '. . . solutions to problems that had previously withstood conventional attacks as well as quick and reliable solutions to problems that had previously yielded comparably effective solutions grudgingly'.

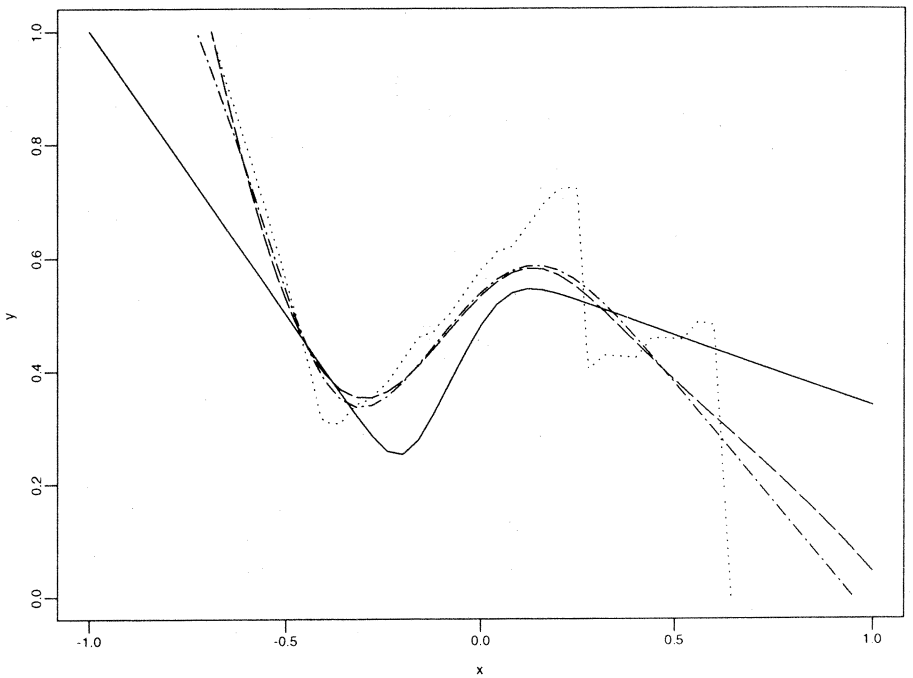

Fig. 4. Boundaries for neural networks with three and six hidden units (note how using weight decay smooths the fitted surface): ———, Bayes rule; · · · · ·, three-node neural net without weight decay; – · –, three-node neural net with weight decay; — —, six-node neural net with weight decay

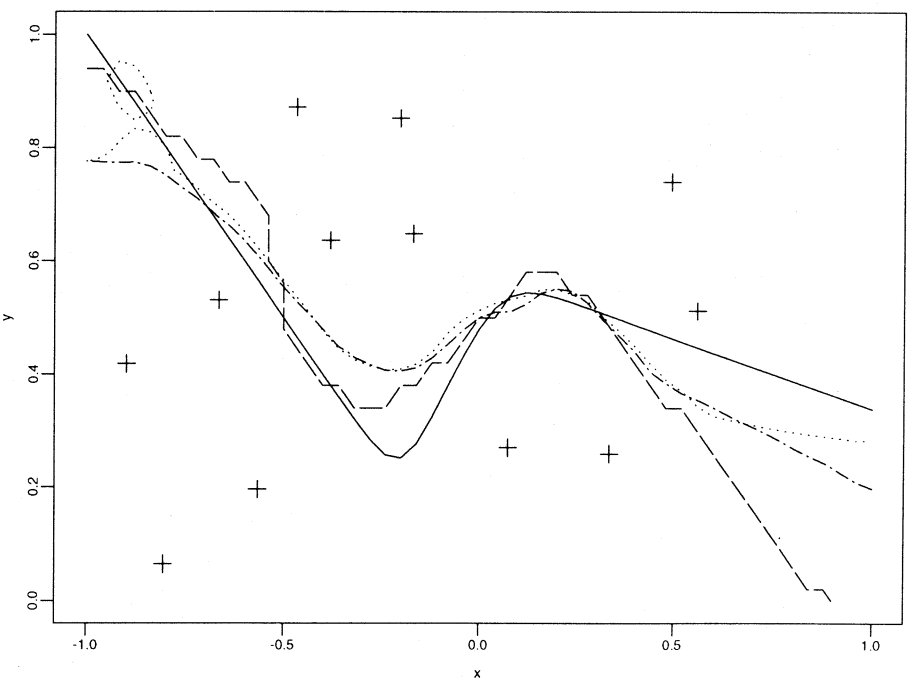

Fig. 5. Boundaries for PPR with two (– · –) and three (· · · · ·) projection terms; also shown are the 12 'code book' vectors chosen for learning vector quantization (+) and its decision boundary (— —) and the Bayes rule (———)

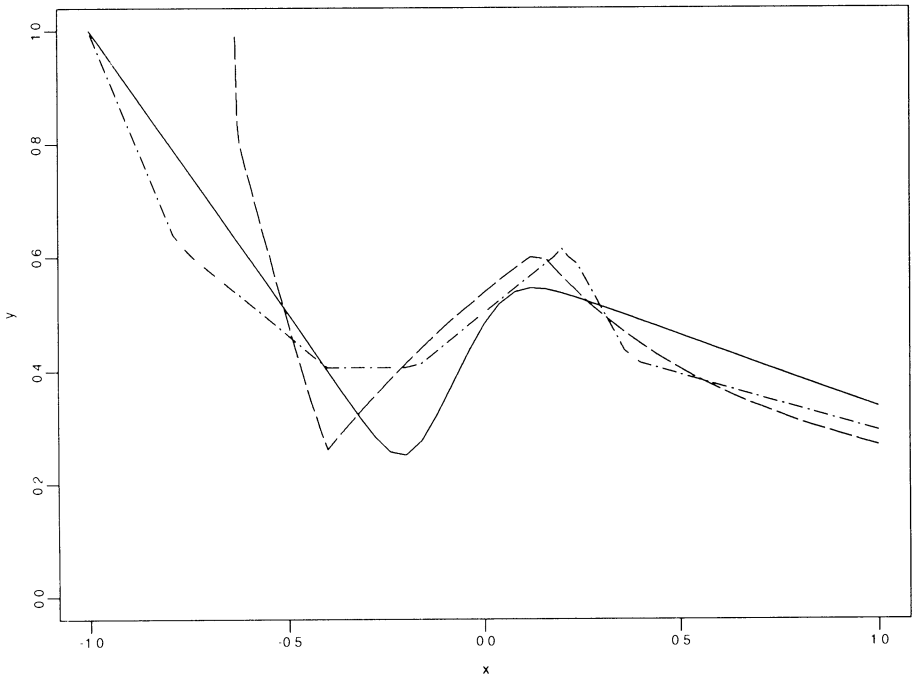

Fig. 6. Boundaries for MARS with degrees 1 (– · –) and 2 (— —) and the Bayes rule (———)

The data are sonar signals bounced off a metal cylinder (a mine, 111 signals) and off a roughly cylindrical rock (97 signals). The signals are energies in each of 60 wave bands. The patterns were ordered in time as the boat moved over the objects, and hence also by angle of incidence. The data were divided into training and test sets of size 104 each, using a cluster analysis to ensure as even matching as possible. (The training set is split 49/55, the test set 62/42, which violates the sampling assumptions of many of the methods.)

Gorman and Sejnowski (1988) fitted a 60–24–2 neural net (coding the outputs for the two classes separately, so this is a fully connected network with 24 units in the hidden layer), which has 1514 parameters, and found an average test set error rate of 10.8%.

The mean responses for the two classes are shown in Fig. 7. This suggests that the responses are smooth, as confirmed by Fig. 8, and so the 60 channels of data are not independent. I averaged them in bands of 3, to make 20 composite channels. The results are given in Table 2, in column (a) for all 60 variables and column (b) for 20 variables. This training and test set are artificially similar, so the results were repeated for training and test sets made up of alternate samples from the data set, shown in columns (c) (60 variables) and (d) (20 variables).

The second set of nearest neighbour results were obtained by rescaling each channel to unit within-group variance on the training set. The results for neural network are averages over five sets of starting weights. Skip-layer connections were included, and weight decay and entropy fitting were used in the training set. The number of weights is large, exceeding the number of examples except for 20 variables and two hidden units. The extraordinary behaviour of the large nets in this example (with 60 variables

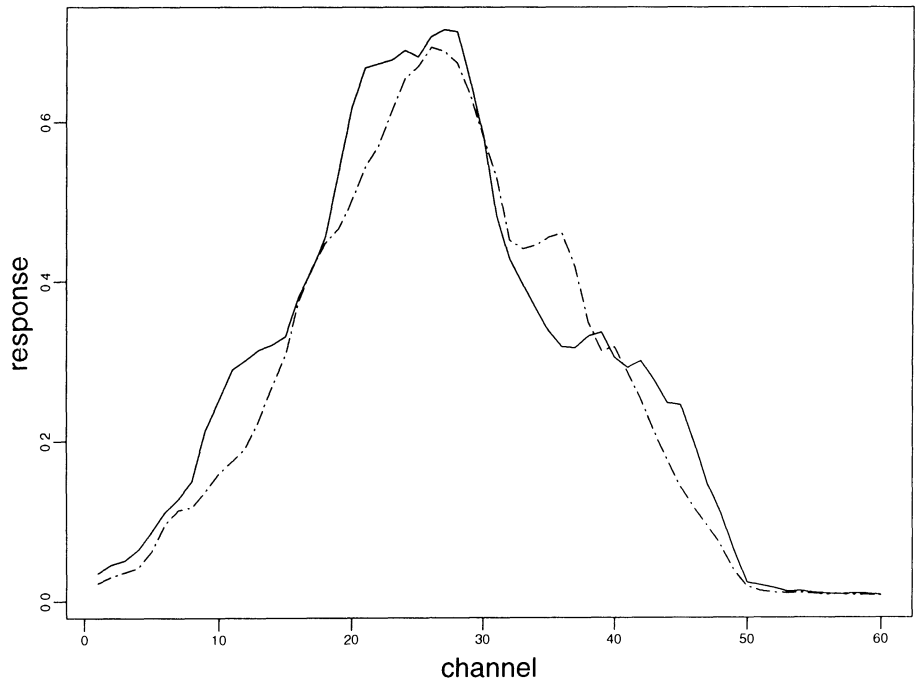

Fig. 7. Mean response for the sonar data: ——, rock; – · –, metal

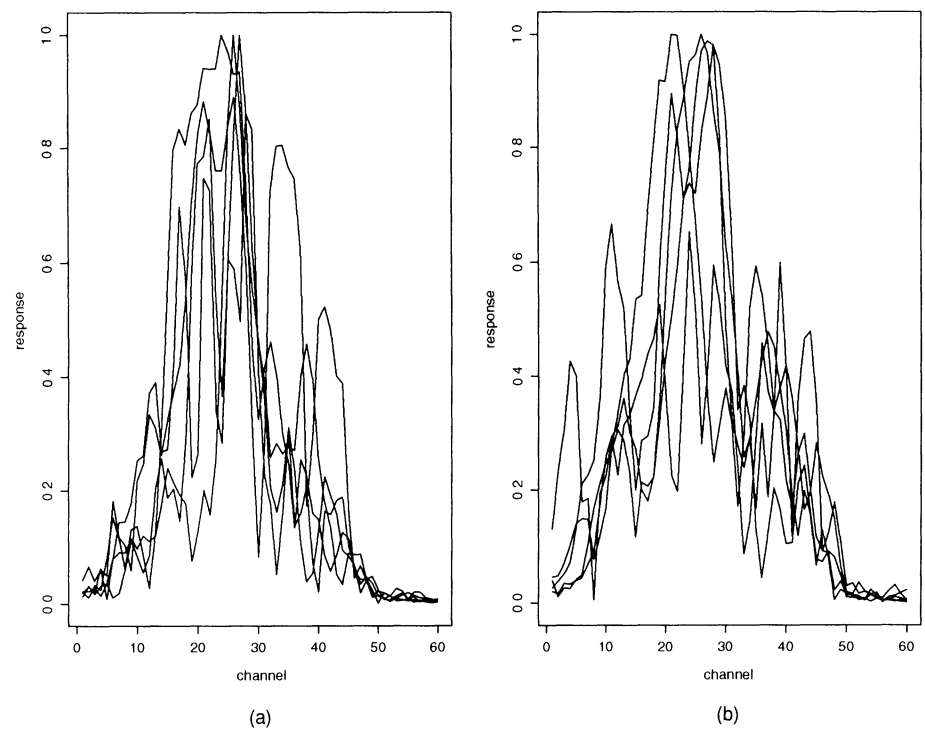

Fig. 8. Five randomly chosen response curves for each class of the sonar data: (a) rock; (b) metal

TABLE 2
Error rates on tests sets of size 104 from the sonar data†

Method	*Errors out of 104*			
	(a)	*(b)*	*(c)*	*(d)*
Linear discriminant	25	20	28	20
Nearest neighbour	9	8	21	17
Nearest neighbour on rescaled variances	4	6	17	20
Neural network with 2 hidden units	17	21	19	23
Neural network with 6 hidden units	12	20	17	20
Neural network with 12 hidden units	11	20	15	19
PPR ($r=2$ or $r=3$)	29	24	23	17
MARS (degree 1)	30	23	21	21
MARS (degree 2)	30	20	26	27
Classification tree	29	21	28	26

†Columns (a) and (c) refer to results based on all 60 channels, columns (b) and (d) to 20 averages of three adjacent channels. Columns (a) and (b) refer to the 'balanced' test set of Gorman and Sejnowski (1988), and columns (c) and (d) to a randomly selected test set.

TABLE 3
Error rates for the sonar data, based on extrapolating from the first 60% of measurements to the last 40%

Method	*Errors out of 84*
Linear discriminant	39
Nearest neighbour	42
1 nearest neighbour on rescaled variances	41
Neural network with 2 hidden units	48, 50, 37, 36, 36
PPR ($r=3$)	36
MARS (degree 1)	36
Classification tree	29

and 12 hidden units there are 805 weights) seems to reflect a need to approximate the nearest neighbour classifier, rather than finding any real structure in the data.

We then used the first 60% of each run of the boat to train, the rest to test, to see whether any real generalization had been achieved (Table 3), which clearly it had not been. The five neural net runs are given in decreasing order of fit to the training data. These results are for 20 variables, but similar results were obtained for all 60 channels. The second problem is more difficult in that it involves a larger degree of extrapolation (in so far as this is pertinent in such high dimensional spaces).

8.3. *Forensic Glass*

The forensic glass data set was collected by B. German on 214 fragments of glass, and taken from Murphy and Aha (1992). Each has a measured refractive index and composition (weight per cent of oxides of sodium, magnesium, aluminium, silicon, potassium, calcium, barium and iron). The fragments were originally classed as seven types, but some are infrequent or absent, and I regrouped them as window float glass (70), window non-float glass (76), vehicle window glass (17) and other (22), omitting headlamp glass which seemed rather distinct in some preliminary linear discriminant

TABLE 4
Error rates for the forensic glass data with four classes†

Method	*Error rate (%)*	
Linear discriminant	41	22
Nearest neighbour	26	17
Neural network with 2 hidden units	38	17
Neural network with 6 hidden units	33	16
Neural network with 6 hidden units and logistic outputs	39	17
Neural network with 6 hidden units via linear discriminant analysis	41	19
PPR ($r=2$ and $r=5$)	40	19
MARS (degree 1)	37	17
MARS (degree 2)	31	19
Classification tree	28	15
Learning vector quantization (11 representatives)	31	17

†The first column of rates refers to all errors, the second to errors excluding confusing window non-float and window float glass. Pairwise comparisons have a 95% confidence interval of half-length about 5%.

analysis. The data set was split randomly, resulting in a training set of size 89 and a test set of size 96.

The results are given in Table 4. The neural network results are averaged over five runs. They show both log-linear (softmax) models, separate logistic models fitted via equation (3) and least square fits with linear output units. The PPR and MARS results used least squares fitting and post-processing by linear discriminant analysis. As some errors are more common and perhaps less serious than others, in the second figure confusion between the two types of window glass is allowed.

There is little to choose between the non-linear methods, most of which show slight but useful gains over linear discrimination. It seems that the Breiman–Ihaka approach is inferior to log-linear models in this example (and in most others of my experience) but results of log-linear PPR and MARS are unavailable because of lack of suitable software.

Figs 9 and 10 show linear and generalized (via MARS) discriminants for this data set.

9. CHALLENGES

Our examples are perhaps a little unrepresentative in that linear methods do moderately well, and the gains by non-linear methods are less spectacular than in, for example, the tsetse fly problem of Ripley (1993). The gains are worthwhile, however, and in our examples achieved for rather modest amounts of computation. It seems to me that the statistical community has not thought enough about how to use (well) large amounts of computation. Most data sets are very expensive to collect relative to both human and computer costs of analysis, and we ought to be thinking how best to spend 24 h of workstation time on classification problems. (Some approaches appear to need years of computer time, including Monte Carlo versions of a full Bayesian analysis in realistic problems.)

However, training data are rarely plentiful and often expensive to classify, so it is important for estimation to be statistically efficient. Here classification has been studied much less than regression, not least because asymptotics are not very relevant,

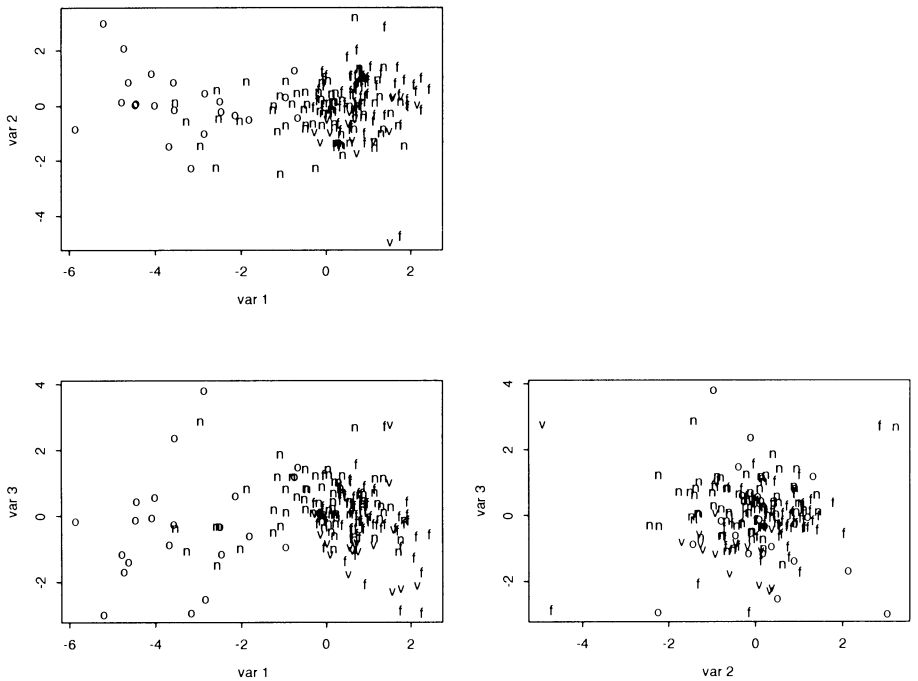

Fig. 9. Discriminant plots for the first three linear discriminants for the forensic glass data: v, vehicle glass; f, window float glass; n, window non-float glass; o, other

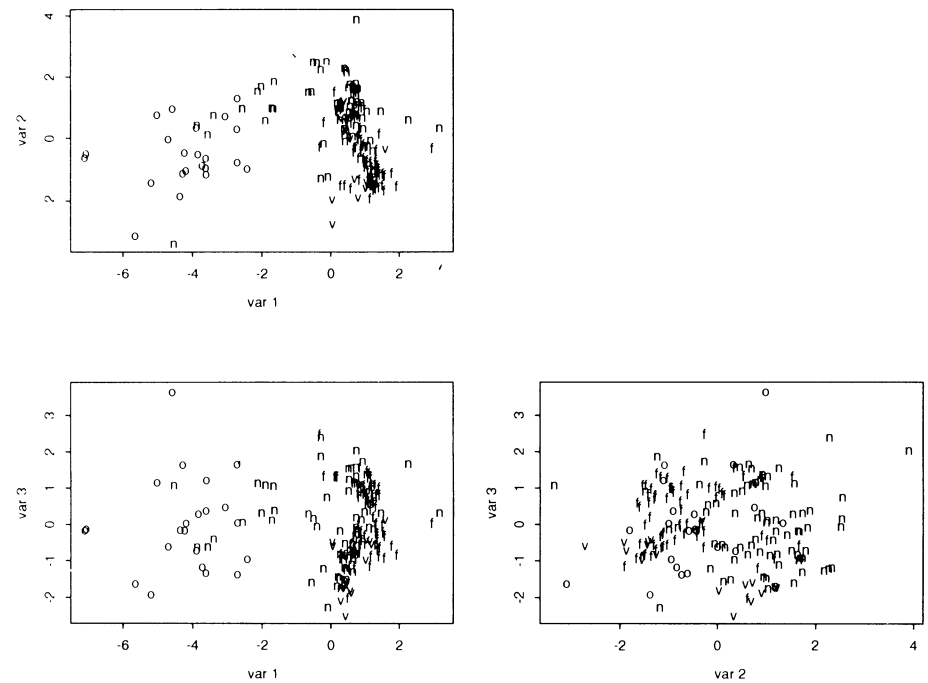

Fig. 10. Generalized discriminant plots from MARS with degree 1: note how the second variable more effectively separates the two types of window glass

and more insight is needed. The value of modelling the whole distribution $p(\mathbf{x}, y)$ rather than $p(y|\mathbf{x})$ alone remains unclear, except where unclassified examples are readily available.

The assessment of performance remains problematical, not least because the training set may not be fully representative of the class of patterns to be classified. For example, in symbol recognition, styles of symbols change with time, and updating methods for the parameters can be very helpful. It is far too common for classifiers to be tuned to look good on the test set, yet tuning non-linear methods is crucial to their success.

What is not in doubt is that there are very many classification problems in which non-linear methods can considerably outperform classical methods, and as automated data collection increases such problems will continue to proliferate. Neurobiologists and psychologists are also looking for insights into the success of humans in solving complex pattern recognition problems apparently effortlessly.

ACKNOWLEDGEMENTS

I am grateful to Andrew Barron, Chris Bishop, Wray Buntine, Jerry Friedman, Trevor Hastie, Nils Hjort, Steve Muggleton, Radford Neal, Rob Tibshirani and Bob Williamson for enlightening conversations and comments on this subject, as well as for pointing me to as yet unpublished work. Ross Ihaka supplied a copy of Breiman and Ihaka (1984), and Rob Tibshirani kindly provided the Hastie–Tibshirani S and Ratfor code for MARS and optimal scoring.

REFERENCES

Abu-Mostafa, Y. S. (1989) The Vapnik–Chervonenkis dimension: information *vs* complexity in learning. *Neural Computn*, **1**, 312–317.

Aitchison, J. and Dunsmore, I. R. (1975) *Statistical Prediction Analysis*. Cambridge: Cambridge University Press.

Amari, S.-I. and Murata, N. (1993) Statistical theory of learning curves under entropic loss criterion. *Neural Computn*, **5**, 140–153.

Amit, D. J. (1989) *Modeling Brain Function*. Cambridge: Cambridge University Press.

Anderson, E. (1935) The irises of the Gaspe Peninsula. *Bull. Am. Iris Soc.*, **59**, 2–5.

Anderson, J. A. (1982) Logistic discrimination. In *Handbook of Statistics 2: Classification, Pattern Recognition and Reduction of Dimensionality* (eds P. R. Krishnaiah and L. Kanal), pp. 169–191. Amsterdam: North-Holland.

Barron, A. R. (1993a) Universal approximation bound for superpositions of a sigmoid function. *IEEE Trans. Inform. Theory*, **39**, 930–945.

——(1993b) Approximation and estimation bounds for artificial neural networks. *Mach. Learn.*, to be published.

Baum, E. B. (1990) The perceptron algorithm is fast for non-malicious distributions. *Neural Computn*, **2**, 248–260.

Baum, E. B. and Haussler, D. (1989) What size net gives valid generalization? *Neural Computn*, **1**, 151–160.

Bishop, C. (1991a) A fast procedure for retraining the multilayer perceptron. *Int. J. Neural Syst.*, **2**, 229–236.

——(1991b) Improving the generalization properties of radial basis function neural networks. *Neural Computn*, **3**, 579–588.

——(1992) Exact calculation of the Hessian matrix for the multilayer perceptron. *Neural Computn*, **4**, 494–501.

——(1993) Curvature-driven smoothing: a learning algorithm for feedforward networks. *IEEE Trans. Neural Netwks*, to be published.

Blumer, A., Ehrenfeucht, A., Haussler, A. and Warmuth, M. (1989) Learnability and the Vapnik–Chervonenkis dimension. *J. Ass. Comput. Mach.*, **36**, 926–965.

Breiman, L. (1991) The Π-method for estimating multivariate functions from noisy data. *Technometrics*, **33**, 125–160.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and Regression Trees*. Monterey: 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.

Brent, R. P. (1991) Fast training algorithms for multilayer neural nets. *IEEE Trans. Neural Netwks*, **2**, 346–354.

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

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

Broomhead, D. S. and Lowe, D. (1988) Multivariable functional interpolation and adaptive networks. *Complex Syst.*, **2**, 321–355.

Buntine, W. L. (1992) Learning classification trees. *Statist. Comput.*, **2**, 63–73.

Buntine, W. L. and Weigend, A. S. (1991) Bayesian back-propagation. *Complex Syst.*, **5**, 603–643.

———(1993) Calculating second derivatives on feed-forward networks. *IEEE Trans. Neural Netwks*, to be published.

Campbell, N. A. and Mahon, R. J. (1974) A multivariate study of variation in two species of rock crab of genus *Leptograpsus*. *Aust. J. Zool.*, **22**, 417–425.

Carroll, R. J. and Pederson, S. (1993) On robustness in the logistic regression model. *J. R. Statist. Soc.* B, **55**, 693–706.

Chan, C. and Bao, J. (1991) On the design of a tree classifier and its application to speech recognition. *Int. J. Pattn Recogn Artif. Intell.*, **5**, 677–692.

Chou, P. A. (1991) Optimal partitioning for classification and regression trees. *IEEE Trans. Pattn Anal. Mach. Intell.*, **13**, 340–354.

Chou, P. A., Lookabaugh, T. and Gray, R. M. (1989) Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Trans. Inform. Theory*, **35**, 299–315.

Ciampi, A., Chang, C.-H., Hogg, S. and McKinney, S. (1987) Recursive partitioning: a versatile method for exploratory data analysis in biostatistics. In *Biostatistics* (eds I. B. McNeil and G. J. Umphrey), pp. 23–50. Dordrecht: Reidel.

Clark, L. A. and Pregibon, D. (1992) Tree-based models. In *Statistical Models in S* (eds J. M. Chambers and T. J. Hastie), pp. 377–419. Pacific Grove: Wadsworth and Brooks/Cole.

Cohn, D. and Tesauro, G. (1992). How tight are the Vapnik–Chervonenkis bounds? *Neural Computn*, **4**, 249–269.

Coomans, D. and Broeckaert, I. (1986) *Potential Pattern Recognition*. Letchworth: Research Studies Press.

Copas, J. B. (1988) Binary regression models for contaminated data (with discussion). *J. R. Statist. Soc.* B, **50**, 225–265.

Crawford, S. L. (1989) Extensions to the CART algorithm. *Int. J. Man–Mach. Stud.*, **31**, 197–217.

Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Math. Control Syst. Sign.*, **2**, 303–314.

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

Diaconis, P. and Shahshahani, M. (1984) On non-linear functions of linear combinations. *SIAM J. Sci. Statist. Comput.*, **5**, 175–191.

Donoho, D. L. and Johnstone, I. M. (1989) Projection-based approximation and a duality with kernel methods. *Ann. Statist.*, **17**, 58–106.

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

Fahlman, S. E. and Lebiere, C. (1990) The cascade-correlation learning architecture. In *Advances in Neural Information Processing Systems 2* (ed. D. S. Touretzky), pp. 524–532. San Mateo: Morgan Kaufmann.

Fayyad, U. M. and Irani, K. B. (1992) On the handling of continuous-valued attributes in decision tree generation. *Mach. Learn.*, **8**, 87–102.

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Ann. Eugen.*, **7**, 179–188.
Fletcher, R. (1987) *Practical Methods of Optimization*. Chichester: Wiley.
Friedman, J. H. (1991) Multivariate adaptive regression splines (with discussion). *Ann. Statist.*, **19**, 1–141.
Friedman, J. H. and Silverman, B. W. (1989) Flexible parsimonious smoothing and additive modelling (with discussion). *Technometrics*, **31**, 3–39.
Friedman, J. H. and Stuetzle, W. (1981) Projection pursuit regression. *J. Am. Statist. Ass.*, **76**, 817–823.
Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition*, 2nd edn. London: Academic Press.
Funahashi, K. (1989) On the approximate realization of continuous mappings by neural networks. *Neural Netwks*, **2**, 183–192.
Gallant, S. L. (1993) *Neural Network Learning and Expert Systems*. Cambridge: Massachusetts Institute of Technology Press.
Gelfand, S. B. and Delp, E. J. (1991) On tree structured classifiers. In *Artificial Neural Networks and Statistical Pattern Recognition: Old and New Connections* (eds I. K. Sethi and A. K. Jain), pp. 51–70. Amsterdam: North-Holland.
Gelfand, S. B. and Mitter, S. K. (1991) Recursive stochastic algorithms for global optimization in $\mathbf{R}^d$. *SIAM J. Control Optimzn*, **29**, 999–1018.
Gelfand, S. B., Ravishankar, C. S. and Delp, E. J. (1991) An iterative growing and pruning algorithm for classification tree design. *IEEE Trans. Pattn Anal. Mach. Intell.*, **13**, 163–174.
Geman, S., Bienenstock, E. and Doursat, R. (1992) Neural networks and the bias/variance dilemma. *Neural Computn*, **4**, 1–58.
Gill, P. E., Murray, W. and Wright, M. H. (1981) *Practical Optimization*. London: Academic Press.
Girosi, F. and Poggio, T. (1990) Networks and the best approximation property. *Biol. Cyb.*, **63**, 169–176.
Gorman, R. P. and Sejnowski, T. J. (1988) Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Netwks*, **1**, 75–89.
Gull, S. F. (1989) Developments in maximum entropy data analysis. In *Maximum Entropy and Bayesian Methods* (ed. J. Skilling), pp. 53–71. Dordrecht: Kluwer.
Guo, H. and Gelfand, S. B. (1992) Classification trees with neural network feature extraction. *IEEE Trans. Neural Netwks*, **3**, 923–933.
Hall, P. J. and Wand, M. P. (1988) On non-parametric discrimination using density differences. *Biometrika*, **75**, 541–547.
Hand, D. J. (1982) *Kernel Discriminant Analysis*. Letchworth: Research Studies Press.
Härdle, W. (1990) *Applied Nonparametric Regression*. Cambridge: Cambridge University Press.
——(1991) *Smoothing Techniques with Implementation in S*. New York: Springer.
Hastie, T., Buja, A. and Tibshirani, R. (1992) Penalized discriminant analysis. To be published.
Hastie, T. J. and Tibshirani, R. J. (1990) *Generalized Additive Models*. London: Chapman and Hall.
Hastie, T., Tibshirani, R. and Buja, A. (1992) Flexible discriminant analysis. To be published.
Hertz, J., Krogh, A. and Palmer, R. G. (1991) *Introduction to the Theory of Neural Computation*. Redwood City: Addison-Wesley.
Hinton, G. E. (1989) Connectionist learning procedures. *Artif. Intell.*, **40**, 185–234.
Hjort, N. L. (1986) Notes on the theory of statistical symbol recognition. *Report 778*. Norwegian Computing Center, Oslo.
Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Netwks*, **2**, 359–366.
Hwang, J.-N., Lay, S.-R., Maechler, M., Martin, D. and Schimert, J. (1993) Regression modeling in back-propagation and projection pursuit learning. *IEEE Trans. Neural Netwks*, to be published.
Hwang, J.-N., Li, D., Maechler, M., Martin, D. and Schimert, J. (1992a) A comparison of projection pursuit and neural network regression modeling. In *Advances in Neural Information Processing Systems 4* (eds J. E. Moody, S. J. Hanson and R. P. Lippmann), pp. 1159–1166. San Mateo: Morgan Kaufmann.
——(1992b) Projection pursuit learning networks for regression. *Engng Appl. Artif. Intell.*, **5**, 193–204.
Jones, L. K. (1987) On a conjecture of Huber concerning the convergence of projection pursuit regression. *Ann. Statist.*, **15**, 880–882.
——(1990) Constructive approximations for neural networks by sigmoidal functions. *Proc. IEEE*, **78**, 1586–1589.
——(1992) A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Ann. Statist.*, **20**, 608–613.

Judd, J. S. (1990) *Neural Network Design and the Complexity of Learning*. Cambridge: Massachusetts Institute of Technology Press.

King, R. D., Muggleton, S., Lewis, R. A. and Sternberg, M. J. E. (1992) Drug design by machine learning: the use of inductive logic programming to model the structure–activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. Natn. Acad. Sci. USA*, **89**, 11322–11326.

Kohonen, T. (1990) The self-organizing map. *Proc. IEEE*, **78**, 1464–1480.

Kůrková, V. (1992) Kolmogorov's theorem and multilayer neural networks. *Neural Netwks*, **5**, 501–506.

Kushner, H. (1987) Asymptotic global behaviour for stochastic approximation and diffusions with slowly decreasing noise effects: global minimization via Monte Carlo. *SIAM J. Control Optimzn*, **47**, 169–185.

Lesaffre, E. and Albert, A. (1989) Partial separation in logistic discrimination. *J. R. Statist. Soc.* B, **51**, 109–116.

Loh, W. and Vanichsetakul, N. (1988) Tree-structured classification via generalized discriminant analysis. *J. Am. Statist. Ass.*, **83**, 715–728.

Macintyre, A. and Sontag, E. D. (1993) Finiteness results for sigmoidal "neural" networks. In *Proc. 25th A. Symp. Theory Computing, San Diego*. To be published.

MacKay, D. J. C. (1992) A practical Bayesian framework for backprop networks. *Neural Computn*, **4**, 448–472.

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

Miller III, W. T., Sutton, R. S. and Werbos, P. J. (eds) (1990) *Neural Networks for Control*. Cambridge: Massachusetts Institute of Technology Press.

Moody, J. E. (1992) The *effective* number of parameters: an analysis of generalization and regularization in non-linear learning systems. In *Advances in Neural Information Processing Systems 4* (eds J. E. Moody, S. J. Hanson and R. P. Lippmann), pp. 847–854. San Mateo: Morgan Kaufmann.

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

Murata, N., Yoshizawa, S. and Amari, S. (1991) A criterion for determining the number of parameters in an artificial neural networks model. In *Artificial Neural Networks* (eds T. Kohonen, K. Mäkisara, O. Simula and J. Kangas), pp. 9–14. Amsterdam: North-Holland.

Murphy, P. M. and Aha, D. W. (1992) *UCI Repository of Machine Learning Databases*. Department of Information and Computer Science, University of California, Irvine. (Available from ics.uci.edu in directory pub/machine-learning-databases.)

Musavi, M. T., Ahmed, W., Chan, K. H., Faris, K. B. and Hummels, D. M. (1992) On the training of radial basis function classifiers. *Neural Netwks*, **5**, 595–603.

Neal, R. (1992) Bayesian training of backpropagation methods by the hybrid Monte Carlo method. *Technical Report CRG-TR-92-1*. Department of Computer Science, University of Toronto, Toronto.

——(1993) Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems 5* (eds C. L. Giles, S. J. Hanson and J. D. Cowan), pp. 475–482. San Mateo: Morgan Kaufmann.

van Ooyen, A. and Nienhuis, B. (1992) Improving the convergence of the back-propagation algorithm. *Neural Netwks*, **5**, 465–471.

Park, J. and Sandberg, I. W. (1991) Universal approximation using radial-basis-function networks. *Neural Computn*, **3**, 246–257.

Poggio, T. and Girosi, F. (1990) Networks for approximation and learning. *Proc. IEEE*, **78**, 1481–1497.

Powell, M. J. D. (1987) Radial basis functions for multivariate interpolation: a review. In *Algorithms for Approximation* (eds J. C. Mason and M. G. Cox), pp. 143–167. Oxford: Clarendon.

Pregibon, D. (1982) Resistant fits for some commonly used logistic models with medical applications. *Biometrics*, **38**, 485–498.

Quinlan, J. R. (1979) Discovering rules by induction from large classes of examples. In *Expert Systems in the Microelectronic Age* (ed. D. Michie). Edinburgh: Edinburgh University Press.

——(1983) Learning efficient classification procedures and their application to chess end-games. In *Machine Learning* (eds R. S. Michalski, J. G. Carbonell and T. M. Mitchell), pp. 463–482. Palo Alto: Tioga.

——(1986) Induction of decision trees. *Mach. Learn.*, **1**, 81–106.

——(1990) Decision trees and decision making. *IEEE Trans. Syst. Man Cyb.*, **20**, 339–346.

——(1993) *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann.

Ripley, B. D. (1993) Statistical aspects of neural networks. In *Networks and Chaos—Statistical and Probabilistic Aspects* (eds O. E. Barndorff-Nielsen, J. L. Jensen and W. S. Kendall), pp. 40–123. London: Chapman and Hall.

Ripley, B. D. and Hjort, N. L. (1994) *Pattern Recognition and Neural Networks—a Statistical Approach*. Cambridge: Cambridge University Press. To be published.

Rissanen, J. (1987) Stochastic complexity (with discussion). *J. R. Statist. Soc.* B, **49**, 223–239, 253–265.

Roeder, K. (1990) Density estimation with confidence sets exemplified by superclusters and voids in galaxies. *J. Am. Statist. Ass.*, **85**, 617–624.

Sethi, I. K. (1990) Entropy nets: from decision trees to neural networks. *Proc. IEEE*, **78**, 1605–1613.

——(1991) Decision tree performance enhancement using an artificial neural network implementation. In *Artificial Neural Networks and Statistical Pattern Recognition* (eds I. K. Sethi and A. K. Jain), pp. 71–88. Amsterdam: North-Holland.

Shanno, D. F. (1990) Recent advances in numerical techniques for large-scale optimization. In *Neural Networks for Control* (eds W. T. Miller III, R. S. Sutton and P. J. Werbos), pp. 171–178. Cambridge: Massachusetts Institute of Technology Press.

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

Spackman, K. A. (1992) Maximum likelihood training of connectionist models: comparison with least-squares back propagation and logistic regression. *Proc. 15th A. Symp. Computer Applications in Medical Care, Nov. 1991*, pp. 285–289. New York: Institute of Electrical and Electronics Engineers.

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

——(1990b) Probabilistic neural networks and the polynomial Adaline as complementary techniques for classification. *IEEE Trans. Neural Netwks*, **1**, 111–121.

Stone, C. J. (1982) Optimal global rates of convergence for nonparametric regression. *Ann. Statist.*, **10**, 1040–1053.

Styblinski, M. A. and Tang, T.-S. (1990) Experiments in non-convex optimization: stochastic approximation and simulated annealing. *Neural Netwks*, **3**, 467–484.

Tråvén, H. G. C. (1991) A neural network approach to statistical pattern classification by "semi-parametric" estimation of probability density functions. *IEEE Trans. Neural Netwks*, **2**, 366–377.

Valiant, L. G. (1984) A theory of the learnable. *Communs Ass. Comput. Mach.*, **27**, 1134–1142.

Vapnik, V. N. (1982) *Estimation of Dependences Based on Empirical Data*. Berlin: Springer.

Wahba, G. (1990) *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Mathematics.

Wallace, C. S. and Freeman, P. R. (1987) Estimation and inference by compact encoding (with discussion). *J. R. Statist. Soc.* B, **49**, 240–265.

Weigend, A. S. and Gershenfeld, N. A. (eds) (1994) *Time Series Prediction: Forecasting the Future and Understanding the Past*. Reading: Addison-Wesley.

Weiss, S. M. and Kulikowski, C. A. (1991) *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*. San Mateo: Morgan Kaufmann.

White, H. (1989a) Learning in artificial neural networks: a statistical perspective. *Neural Computn*, **1**, 425–464.

——(1989b) Some asymptotic results for learning in single hidden layer feedforward networks. *J. Am. Statist. Ass.*, **84**, 1008–1013.

——(1992) *Artificial Neural Networks: Approximation and Learning Theory*. Oxford: Basil Blackwell.

Zhao, Y. and Atkeson, C. G. (1992) Some approximation properties of projection pursuit learning networks. In *Advances in Neural Information Processing Systems 4* (eds J. E. Moody, S. J. Hanson and R. P. Lippmann), pp. 936–943. San Mateo: Morgan Kaufmann.

DISCUSSION OF THE PAPER BY RIPLEY

P. Whittle (University of Cambridge): I used to cry in the wilderness, as I felt, voicing the view that statisticians had narrowed down their tradition and were not involving themselves with exciting new developments which were essentially statistical in nature although of a non-classical cast, e.g. expert systems, latent variable models, image restoration, tomography and, most recently, neural nets. I do

not believe that I was heeded to any extent; rather, statistical good sense simply reasserted itself, and I have been sufficiently overtaken by events that I am pleased on the one hand and definitely quieter on the other.

Neural nets are very much a case in point, since not only did neural net enthusiasts claim to have bypassed statistics in some senses; they rather spoiled their claim by making a point of the fact that neural nets could reproduce classical multivariate techniques such as those of principal components and discriminant analysis. It is good, then, that statisticians assess the situation for themselves in quite an active way. Professor Titterington has written a couple of survey articles of this character and, as we have now seen, Professor Ripley has also developed his own view, a strongly algorithmic one.

Neural nets (or, more properly, artificial neural nets) are sufficiently new as an enthusiasm that it is still reasonable to ask what they are, what they are meant to do and what they can do. The original hope was, of course, that they would mimic and throw light on the operation of natural neural nets, and operate with some of their adaptability and power. The naïve hope was that they would provide the universal and ultimate package: that, exposed to sufficiently much data from a given source, they would not merely fit parameters but find a model and then indeed explain it, leaving the human investigator only with the more honorific task of writing the paper and collecting the credit. My view is that the interest lies in seeing how far a simple-minded improvement algorithm could lift the system, if left to run. Could it really lift the system to a higher level of sophistication—the ultimate bootstrap? Could this blind ratchet of a mechanism arrive at something more enlightened and reveal that enlightenment? Even if it could not, then the permanent installation of such an improvement rule seemed at least to provide a degree of in-built adaptability, an in-built protection against changing conditions and against natural deterioration of the system. Here I am thinking of the system as something which realizes an input–output relationship, the input being the current environment and the output the best response, in some sense, to that environment.

However, another view seems now to have developed: that this realization of an input–output relationship is just the fitting of a rather general non-linear relationship or model. The neural net technique is then regarded as just one of many possible fitting algorithms, one which offers the particular features of an economical representation of the non-linear functions (in terms of connection constants) and the availability of an existing improvement algorithm (the back-propagation algorithm), in other words, a kind of general purpose statistical fitting procedure, hypothesis free and so certainly free of any notion of significance testing. It is this use which Professor Ripley explores, surveys very capably and illustrates very effectively. He certainly regards neural nets as just one of the class of what he has termed 'flexible non-linear approaches to classification'. He reviews the various classification algorithms or representations and sees that they might be trimmed, improved or made cleverer for particular purposes, and how they might then be realized by a neural network. This is a perfectly proper point of view. However, it abandons completely the early motivation for artificial neural networks, which was precisely that the improvement algorithm *should* be simple minded in its operation: that cleverness should not be supplied, but should evolve. The key question was, of course, whether it *would* evolve.

Professor Ripley in fact discusses this hope in Section 7, when he discusses generalization; the capacity of a network to cope with a larger task than that for which its training data might have prepared it. He quotes Judd's discouraging theoretical conclusion, that 'Networks cannot generalize', but then quotes investigations which seem to show better empirical performance than some theory might predict.

Whichever view one takes, a point that should be made is that the standard improvement step, the back-propagation algorithm, has its limitations. If the problem exceeds a certain size, as measured by the dimension of its input, then back-propagation simply chokes and refuses to work. It is at this point that a fundamental advance is needed, to discover an improvement mechanism which is more effective, while still 'natural' and 'simple minded'. Another point is stochasticity of the network itself. Professor Ripley would presumably not want any; for other purposes, it may play a positive role.

Well, the proof of the pudding is in the eating. For Professor Ripley 'eating' consists in finding a flexible and powerful algorithm. For me, the meal may never arrive, but the anticipation is that a large stochastic system, equipped with some kind of adaptation mechanism and exposed to a possibly complex but basically consistent environment, can show behaviour beyond our present understanding.

Jim Kay (University of Stirling): Professor Ripley has inevitably left some gaps in his presentation and I wish to discuss one of them, namely, the business of artificial neural networks for unsupervised feature discovery. In contrast with this paper, in unsupervised learning the correct 'type' is not available: in statistical terms this corresponds to cluster analysis or latent structure analysis rather than

discrimination. Cognitive scientists, who wish to understand how the brain does statistics, view such networks as being important as they attempt to build computational models intended to mimic known neurophysiological and neurobiological aspects of brain function; some of them think that statisticians can help!

Neural networks have been developed for, among other things, the extraction of principal components (Oja, 1989; Sanger, 1989; Hornick and Kuan, 1992; Xu and Yuille, 1992), feature discovery using competitive learning (Rumelhart and Zipser, 1986; Foldiak, 1990), feature discovery using exploratory projection pursuit (Intrator, 1990; Intrator and Cooper, 1992; Intrator and Gold, 1993) and feature discovery using information theoretic objective functions (Linsker, 1988, 1992; Becker, 1991; Becker and Hinton, 1992; Galland and Hinton, 1989).

The work of Intrator and Cooper (1992) is particularly interesting because

(a) it comes with sound theoretical analysis,
(b) the properties of the network mimic physiological phenomena obtained in experiments on kittens—so much so that the Intrator and Cooper conjecture that a neuron might be performing the statistical task of feature extraction—and
(c) the learning rules in their approach have the same form as the famous Bienenstock–Cooper–Munro (BCM) (Bienenstock *et al.*, 1982) rule which has been shown to have some biological plausibility.

The work of Linsker (in particular Linsker (1988)) is fascinating. He performed some experiments using a multilayered linear network. In each layer, a unit received input from only a subset of the units in the previous layer (its receptive field) and the receptive fields of two nearest neighbours overlapped. He introduced random noise into the network and found that the network weights 'learned' to encode feature detectors such as bar and orientation detectors and centre-surround cells—such detectors have been found in the primary visual cortex of some mammals. The aim of his network was to maximize the transmission of information through the net by maximizing the variance of the values on the output units. On the basis of these experiments he proposed his infomax principle, according to which the weights should be learned to maximize the average mutual information between the inputs and outputs. This objective function has the form

$$I(\mathbf{x};\ \mathbf{y}) = H(\mathbf{y}) - H(\mathbf{y} \mid \mathbf{x}).$$

Here H denotes the Shannon entropy and $\mathbf{y}$ and $\mathbf{x}$ denote respectively the outputs from, and inputs to, the network. In statistics this function has been termed the expected gain of information provided by an experiment (Lindley, 1956) and it was used as a principle of feature selection by Aitchison and Kay (1975). The work of Hinton and co-workers also used average mutual information as an objective function to maximize the agreement between two output units in a network, the idea being that an output 'supervises' its neighbour's output and vice versa—a form of mutual self-supervision at the network level. More recently these ideas have been extended, at Stirling, in networks intended to promote the contextual guidance of learning and processing. The aim here is for the network to extract those features of the input receptive fields that are predictably related to the current context. These networks employ a novel activation function, they have learning rules that are mathematically similar to the BCM rule while having quite a different dynamic threshold and they could be used for either supervised or unsupervised learning or indeed in hybrid forms involving both types of learning.

It is clear that Professor Ripley, in a cluster of connected papers including this paper, has made a useful contribution to our understanding of statistical aspects of neural networks for classification and I am very happy to second the vote of thanks.

The vote of thanks was passed by acclamation.

David J. Hand (The Open University, Milton Keynes): I would like to commend Professor Ripley on an impressive *tour de force*. He has covered a large and complex area and yet has managed to highlight most of the important issues. He also makes some vitally important remarks almost as throw-away lines. I particularly endorse his comments about the overriding importance of selecting the right features and about the two distinct purposes in learning a classifier. This distinction can have consequences for the way that the task of constructing the classifier is undertaken.

There was considerable work in pattern recognition in the 1960s and 1970s on feature extraction—trying to find those transformations of the raw variables which contained the information relevant to the classification. This effort tended to impose the view that classification was a two-stage process,

with feature extraction followed by classification. One of the beneficial consequences of work on neural networks is that the process is now more readily seen in its true light as a single complex mapping from the space of measurements to the class space. The author's paper makes this very clear.

However, there is one area to which I think the author has given insufficient attention, namely the assessment of classification methods. As he says, performance is typically measured by the misclassification rate. But a simple count of errors leaves much unsaid. For example, a study of whether the classifiers are misclassifying the same points could lead to improved methods. Moreover, a simple error count can be improved by shrinking the individual class predictions away from 0 and 1. Certainly, if the shrinking is towards the classifiers' estimates of the class membership probabilities then this may introduce a bias which depends on the classifier used (as the author says about methods based on the training set) but some simple standard procedure could be adopted.

Finally I wanted to criticize the author for making the mistake made by almost all researchers in this area: using the standard errors of the rates as the basis of comparisons. These standard errors do not give the relevant error bounds. The test sets are common across classifiers and so are matched. However, the proofs I saw differed from the verbal presentation of the paper, and in the latter he pointed out the need to take the correlation into account. So, instead of criticizing him, I would like to commend him.

Lionel Tarassenko (University of Oxford): Radial basis function (RBF) networks are mentioned in passing by Professor Ripley. The main advantage of these networks is usually perceived to be the much lower training times in comparison with multilayer perceptrons (MLPs) and error back-propagation. Training an RBF classifier usually consists of two phases: the position of the centres in input space is first determined by using unsupervised clustering methods such as the adaptive K-means procedure; the widths are then set by using any number of heuristics. In the second phase, the hidden layer representation becomes the input to the second layer which is trained independently. Since this is now a linear optimization task, it can be performed by using matrix pseudoinverse techniques or the LMS algorithm.

This two-phase approach is very fast but classification results tend to be slightly worse than with an MLP. Unsupervised clustering techniques set the free parameters of the first layer purely according to the distribution of the training data in input space: any class information is ignored. If class labels *are* used and the centre positions are adjusted by using gradient descent (error back-propagation), it can be shown that the classification performance of an RBF network is almost identical with that of an MLP. The improvement in training times, however, no longer obtains.

RBF networks offer other significant advantages, if the hidden layer representation is viewed as a Gaussian mixture model. There are classification problems for which we do not know the exact number of classes. For example, in the analysis of X-ray mammograms, we know that there are broadly four types of tissue, namely fat, fibre, cyst and carcinoma, but we do not know how many types of fibrous tissues or cysts exist. The Gaussian mixture model can be used to estimate the number of subclusters in the training data and from this we can decide how best to construct the second layer. Note that, when moving to classification, more than one kernel per subcluster will be required for optimal performance. The mixture model, or hidden layer of the RBF network, has one final advantage: it can be used as a *novelty detector* on test patterns, indicating whether the network is interpolating, as it should be, or extrapolating, in which case no confidence can be attached to the output classification. It can thus be argued that the hidden layer representation of an RBF network is much more powerful than that of an MLP.

Philip J. Brown (University of Liverpool): My interest in this paper stems from experience with discrimination in non-standard situations. Under the Science and Engineering Research Council's Complex Stochastic System initiative, we (with Dr Tom Fearn at University College London) are looking at 'singular discrimination'. The characteristics of problems are

(a) a high dimension (perhaps 1000 variables, the absorbances making up a spectral near infra-red curve),
(b) moderate to many groups (2–100 classes),
(c) a relatively small training set (50–1000 observations) and
(d) covariates defining experimental conditions.

An example is the classification of orange juices (Evans *et al.*, 1993). I have been working more with pharmaceutical examples.

The statistical approach to modelling, teasing out and using the uncertainties involved, selecting regions of the spectrum which are robust to varying experimental conditions, together with a rich range of models, can offer fast and accurate pattern recognition. One particularly appealing class of models arises from Bayesian model choice. This allows uncertainty in the model as well as the parameters of a particular parametric model. If M_i is a model that specifies a subset of the (normal) populations to have equal covariances, then compromise methods between linear and quadratic discrimination are possible; see Smith and Spiegelhalter (1981). The predictive distribution used in discrimination is

$$p(x|\text{data}) = \sum_{i=1}^{m} p(x|\text{data}, M_i)\, p(M_i|\text{data})$$

where m is the number of models. Friedman (1989) offered a compromise that is similar in spirit.

One point worth emphasizing, whatever the approach, is that the methodology should be alert to the possibility that the object presented does not belong to any of the classes in the training data.

There is some value in subsuming discrimination under the calibration paradigm (see Brown (1993)), if only to emphasize that how the data arose (random or controlled training data) has bearing on the method of analysis. Also scoring y to be predicted by x (Breiman and Ihaka, 1984) arises naturally from the comparison of the two regressions, x on y, and y on x, used to predict y (Brown, 1982, 1993).

D. M. Titterington (University of Glasgow): It is easy to be struck by the greater variety of methods for classification described here than we used (Titterington *et al.*, 1981) in what we then felt was a wide-ranging comparison of methods for the prognosis of outcome after severe head injury. Then there was no mention of neural networks, projection pursuit, multivariate adaptive regression splines; no S-PLUS code for classification and regression trees. A general conclusion of ours was that simpler methods, such as linear discriminant analysis and 'independent Bayes', did surprisingly well, at least on that data set.

Our problem was on a very small scale, and a main message from this paper is that we need to sort out what to do with very large problems. Perhaps often familiar statistical approaches will do well. Kanal (1993) remarks thus in his overview of his subject, but he then outlines a classifier of radar cross-sections which is very complicated, being a hybrid network of 69 component neural networks, 34 of which are multilayer perceptrons.

The scale of some applied problems may be discouraging to statisticians as their general methodology, based on modelling and/or asymptotics, may simply be inapplicable. The predictive Bayes approach of Section 6.4 is difficult to beat, aesthetically, and advances have been made in neural network contexts, but it still seems a long way from general practicability. Many large-scale pattern classification exercises still involve rules constructed with the help of many bits of clever but *ad hoc* tuning and tweaking, to achieve the primary aim, of dealing with that particular application.

In spite of this, there are important general areas for more systematic work, including the development of Bayes-like regularization methods for highly parameterized problems (Geman *et al.*, 1992), further investigation of the properties of universal approximators, along the lines of work by Barron and others, stepwise construction of feed-forward networks (Breiman, 1994) and reliable assessment of generalization ability.

It is natural to comment also on the wider question of statistical interest in research into artificial neural networks (ANNs). In the paper, 'neural networks' usually means 'feed-forward networks with logistic activation functions'. This represents only an admittedly important subclass of artificial neural networks. The complete interface between ANN research and statistics is rather wider, as the author himself describes elsewhere (Ripley, 1993). In our own contribution (Cheng and Titterington, 1994), we were gratified that, in the subsequent discussion, in which Professor Ripley kindly participated, there was, in spite of doubts about the statistical credentials of some of the ANN literature, unanimous agreement that important problems are being addressed, some of them related to real artificial intelligence issues, and that it is vital to *both* communities that statisticians should be deeply involved.

Charles Taylor (University of Leeds): I would like to congratulate Professor Ripley for this insight into the relationships between different classification methods. A comparison of methods, such as that given in Section 8, is often difficult to interpret. Observed differences in goodness of result can arise from

(a) different suitabilities of the basic methods for given data sets,
(b) different sophistications of default procedures for parameter settings,

(c) different sophistication of the program user in the selection of options and tuning of parameters and
(d) the occurrence and effectiveness of preprocessing of the data by the user.

The stronger a program is in respect of (b), then the better the buffer against shortcomings in (c). Alternatively, if there are no options to select or parameters to tune, then item (c) is not important. In the examples described in this paper, the tests were all carried out by one person, so at least some of the variability has been eliminated.

Both the nearest neighbour and kernel density methods are plagued by difficulties in a suitable choice of metric. A related point is that both of these methods can be substantially improved by judicious choice of variables. I have seen real life examples in which the error rate is halved (e.g. from 8.8% to 3.1% on a test set of 5000 examples). Similar improvements can be achieved by transforming the data; for example 'sphering' the data often gives substantial improvements over the standard independent multivariate normal kernel, although this method appears not to be robust. Other examples can be found in Michie *et al.* (1994). Do not such considerations often swamp the minor differences that may exist in error rates from one algorithm to the next?

This leads to another issue of the importance of criteria other than the error rate—such as central processor unit time, memory, comprehensibility to the user and general ease of use of the algorithm. These factors also outweigh minor differences in accuracy for many users.

Walter R. Gilks (Medical Research Council Biostatistics Unit, Cambridge): I am slightly dismayed at the realization that neural networkers are tackling problems that look like statistical problems. However, they are not quite doing what we would or should be doing. Typically, statisticians have adopted a kind of automatic approach to classification in which no real modelling of the data is involved. In this domain neural networks can excel. When the data are less plentiful and have an intricate structure engendered through complexities in the underlying biology or through the study design, it behoves us to *model*.

I have recently been involved in a large international workshop collecting data for a cluster analysis of monoclonal antibodies (Schlossman *et al.*, 1994). Clusters of monoclonals correspond to cell surface antigens. For example, the CD4 antigen, familiar to many statisticians who have worked with data on acquired immune deficiency syndrome, was identified in this way. The cluster analysis needs to take into account many things, including differences between laboratories and assays, and affinity differences between monoclonals recognizing the same antigen. We also know something about the measurement process, in particular the influence of background reactivity. It is possible to bring all these ingredients into large graphical models incorporating an element of clustering (Gilks *et al.*, 1989). Ultimately clusters are confirmed by further laboratory investigations, and the data can then be used for the classification of monoclonals not yet included. At the classification stage a directed acyclic graphical model might be constructed as in Fig. 11.

β denotes cluster-specific parameters; λ denotes laboratory-specific parameters; μ and η denote hyperparameters; x denotes features; y denotes classifiers; squares denote observations; circles denote

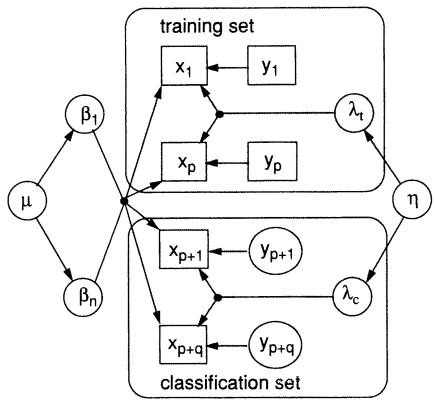

Fig. 11

unknown quantities and missing data; arrows denote conditional independence assumptions. To keep Fig. 11 uncluttered, only clusters 1 and n and monoclonals 1, p, $p+1$ and $p+q$ are shown. The classification parameters y determine which of the cluster parameters β apply, and the laboratory parameters λ_t and λ_c allow for different sets of laboratories for the training and classification data. Thus the classification data are not necessarily exchangeable with the training data, although the training data may provide information about the hyperparameters η, which in turn provide information about λ_c. Allowing for between-laboratory variation is important, especially since the aim of the analysis is to provide *posterior distributions* for the unknown classifiers $y_{p+1}, \ldots, y_{p+q}$, not merely point estimates. Many elaborations on this basic framework are possible. Estimation of the model would probably be through Markov chain Monte Carlo methods although the computational burden can be enormous.

This approach places great emphasis on understanding the processes underlying the data and parsimoniously modelling them, whereas the neural network approach places great emphasis on avoiding substantively meaningful parametric assumptions. There is room for both.

Frank Critchley (University of Birmingham): It is a pleasure to welcome an important statistical contribution to a major emerging area. There are three aspects to my contribution: a question, a note on cognate work and remarks on an assumption.

(a) What can be said, practically or theoretically, about the trade-off between increasing the number of nodes in each of a fixed number of layers and increasing the number of layers each with a fixed number of nodes? For example, when do $n_1 n_2$ nodes in a single layer perform better or worse than n_1 nodes in each of n_2 layers?
(b) Cook (1993a, b) and Cook and Wetzel (1993) are actively exploring the idea of graphical regression. Assuming that (after appropriate transformation) the predictors are elliptically contoured buys the freedom to explore graphically *any* distribution of the dependent variable conditional on them. Very recent collaborative work suggests that extension to a multivariate dependent variable should be feasible.
(c) It appears that, especially when $K>2$, benefit may flow from considering geometries other than a fixed Euclidean geometry. Locally varying metrics are one possibility. The important differential geometrical approach of Amari (1985, 1993) can be difficult for statisticians to access. Critchley *et al.* (1993) have developed an elementary account of Amari's geometry which will be presented shortly at a seminar at Imperial College, London.

Once again it is a pleasure to thank the speaker for a very stimulating paper.

A. J. Mayne (Milton Keynes): I should like to raise two questions. First, has any work been done about recognition of evolving patterns, i.e. where the types of thing that we are trying to classify are no longer static but gradually evolve, even though not much is known about the way in which they evolve? There might perhaps be scope for some methodology to find out what sort of evolution there is. I do not know whether or not this has yet been addressed.

Secondly, instead of, say, considering just one type of neural network, several neural networks might perhaps be set on to the task. The nature of these neural networks might itself be characterized by various parameters. In this particular case, there is a higher order problem and we now try to find what will be the best assignments that can be given to the parameters for this configuration of neural networks to obtain better results.

I must admit that I have not done any work on either of these aspects and am just suggesting them as possibilities.

Grace Wahba (University of Wisconsin, Madison): We heartily thank Professor Ripley for this and his other illuminating contributions to greater understanding and cross-fertilization between the statistical communities and the supervised machine learning communities. Both groups have good things to learn from each other.

We describe smoothing spline analysis-of-variance (ANOVA) maps, which contain the popular additive smoothing spline models as well as parametric GLIM models as special cases and provide an additional, fairly structured way of including interactions. Considering the $K=2$ class case with a representative training set, and letting $f(\mathbf{x}) = \log\{p(1 \mid \mathbf{x})/p(0 \mid \mathbf{x})\}$, $\mathbf{x} = (x_1, \ldots, x_d)$, an estimate $\hat{f}$ of f may be

obtained by assuming a spline ANOVA decomposition of the form $\hat{f}(\mathbf{x}) = \mu + \Sigma_* \hat{f}_\alpha(x_\alpha) + \Sigma_{**} \hat{f}_{\alpha\beta}(x_\alpha, x_\beta) + \ldots$ and choosing $\hat{f}$ as the minimizer of log-likelihood$\{Y_i, X_i | f\} + \Sigma_* \lambda_\alpha J_\alpha(f_\alpha) + \Sigma_{**} \lambda_{\alpha\beta} J_{\alpha\beta}(f_{\alpha\beta}) + \ldots$; here the J_α, $J_{\alpha\beta}$ etc. are spline-like penalty functionals when x_α are continuous variables. The selection of which terms $(_*)$, $(_{**})$ etc. in the sums to include in the ANOVA model may be made by setting aside a fraction of the data for model selection (given a good set of smoothing parameters λ_α and $\lambda_{\alpha\beta}$ for each model under consideration), and a generalized cross-validation type of proxy for the Kullback–Leibler information distance may be used to choose the smoothing parameters (within the training set). As with most of the methods discussed, there are still plenty of open questions on the best way to choose the model plus smoothing parameters. This kind of spline ANOVA model is, or can be, relatively highly structured compared with the other map families considered in Section 3. It allows the possibility of building an *interpretable* structure for the map, particularly with respect to demographic medical data. (How does my risk of outcome 1 vary if I change my risk factor x_α by so much?) At the same time these models demand more input from the user than the maps considered in Section 3, from the point of view of requiring the user to limit *a priori* the possible structures to be considered when d is large, possibly by a preliminary selection procedure. We have recently implemented an example in Wahba *et al.* (1993) (available from ftp.stat.wisc.edu in pub/wahba/soft-class.ps.Z).

Stephen P. Luttrell (Defence Research Agency, Malvern): Perhaps the title of the paper should be 'A certain kind of neural network and related methods for classification', because Professor Ripley does not discuss unsupervised neural network techniques which are widely used for cluster analysis (and other) problems. To be specific, the humble Euclidean error minimizing vector quantizer (or k-means classifier) may be interpreted as a very simple 'winner takes all' unsupervised network, and it may be generalized to produce various other networks.

For instance, the Kohonen map is a type of vector quantizer in which class nodes other than the winning node acquire a non-zero activity by leakage of the winning node's activity onto its neighbours. This coupling causes the class reference vectors of neighbouring nodes to become correlated to form a Kohonen map. If the neighbourhood is two dimensional then the reference vectors self-organize into a two-dimensional sheet that folds up to space-fill those regions of the higher dimensional input space that are populated by training vectors.

The most important side-effect of this self-organization is to force the inverse mapping from class label to class reference vector to be a smooth function of the class label, a property that the standard vector quantizer does not enjoy. This has several desirable consequences.

(a) The pattern of node activity is easier to interpret visually, especially when a two-dimensional neighbourhood is used. This trick is frequently used to visualize high dimensional signal distributions by mapping them onto a two-dimensional Kohonen sheet.
(b) The network may be cascaded to form various kinds of multilayer vector quantizer network, which trade off quantization accuracy against speed of encoding. See, for instance, Luttrell (1989). The basic trick used in multilayer vector quantizers is to simplify the overall function $f(\mathbf{x})$ that maps from input vector to class label thus: $f(\mathbf{x}_1, \mathbf{x}_2) = f_2(f_{11}(\mathbf{x}_1), f_{12}(\mathbf{x}_2))$, where $\mathbf{x}$ is partitioned into two subspaces as $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$. The individual functions $f_{11}(\mathbf{x}_1)$, $f_{12}(\mathbf{x}_2)$ and $f_2(f_{11}, f_{12})$ may all be implemented as vector quantizers, but $f_{11}(\mathbf{x}_1)$ and $f_{12}(\mathbf{x}_2)$ *must* be smoothly invertible (as a Kohonen map is) if their outputs are to be fed into $f_2(f_{11}, f_{12})$. This approach generalizes to an arbitrarily deep nesting of self-organized vector quantizers.

The following contributions were received in writing after the meeting.

A. J. Baczkowski and K. V. Mardia (University of Leeds): A neural net method called a cerebellar model articulation controller (CMAC) (Miller *et al.*, 1987) classifier, which can be regarded as a kernel classifier (Lippmann, 1989), has found popularity with some industrial organizations. We came across this as a method to classify cirrus clouds from Meteosat images.

Specialist meteorologists had stored the images in pure classes: homogeneous, patchy and streaky, although mixed classes are most common. As expected, anisotropic variograms could extract some textural features, but we were originally presented with about 800 texture variables taken from results on co-occurrence matrices (Haralick, 1979), grey level difference vectors (Weszka *et al.*, 1976) and sum and difference histograms (Unser, 1986). The aim was to reduce the number of variables to about 20.

The final set of selected variables was ranked according to their discrimination power. Using different numbers of these ranked variables we have compared discriminant analysis with some neural networks. Surprisingly, in all cases, the learning vector quantization network performed slightly worse than either linear or quadratic discriminant analysis, with classification accuracies in the region 66–72% compared with 64–82% for discriminant analysis on the test set. Of course, the variables which are important in one method are not necessarily useful for another method. Some of these computations were carried out by our student Ms Xiaojuan Feng. The CMAC, operated by the company itself, is claimed to give accuracies in the range 80–85% on the test set.

Have there been any comparative studies of CMAC *versus* other neural networks? Are there any studies of neural nets operating on mixed pixels? Have there been any attempts to extend non-linear discriminant analysis to handling fuzzy classes? We will very much appreciate your comments.

Leo Breiman (University of California, Berkeley): Professor Ripley has presented a most welcome paper—a clear, sensible and thoughtful discussion of major issues revolving around neural nets in particular and classification in general.

Almost by accident, the neural network concepts have stimulated a resurgence of interest in pattern recognition. Unfortunately, most of this has taken place outside statistics and involved young computer scientists, physical scientists and engineers (mainly quite smart and energetic). They are working hard and making progress on problems such as speech and handwritten character recognition, involving high dimensional and large databases.

They are learning that, although neural nets are a good multipurpose tool, they are not a panacea. For instance, at the last Snowbird conference on neural networks in computing (1993) it was reported that the lowest misclassification rate to date on AT&T's handwritten digit database was achieved by a nearest neighbour method by using a smart metric.

Much unwritten folklore is used in running the neural net algorithm. There are various home-brew schemes for figuring out how many nodes to use in the hidden layer. The advice is to never run until you hit the minimum—'that would be overfitting the data'. Instead use devices such as the device mentioned by Ripley. Other practitioners say that they never train to 0, 1 targets, but to more moderate values such as 0.2, 0.8.

There are other aspects of neural nets that puzzle me. For instance, almost none of the neural net people seem to worry about landing in local minima. But it worries me. Is it a problem, and, if not, why not? Is it true that all the local minima give classifiers achieving about the same misclassification rate?: if so, why? Good insight into behaviour on finite data sets does not seem to be available.

Comparing classifiers on a few data sets is problematical. For instance, my experience on the relative efficacy of regression-type *versus* log-linear-type classifiers is the opposite of Ripley's. What classification scheme is best changes from one data context to another. Usually, neural nets, nearest neighbours and classification trees do well, but classical linear methods are more easily thwarted.

Finally, we note that Ripley's neural net software is installed on our departmental machines and express our gratitude to him for making it available publicly.

Wray Buntine (Research Institute for Advanced Computer Science, Moffett Field): In this wide-ranging review, it is difficult to discern neural networks from the modern statistical methods. Nevertheless, what are the positive and constructive aspects that we can learn from neural networks? First, they are claimed to be accessible to the novice, as demonstrated by their widespread use. Second, they handle complex and varied problems by being computer intensive and network based. Another expanding area in the broader field of data analysis—but in this case co-developed in statistics, artificial intelligence and the decision sciences—is graphical models (Whittaker, 1990; Spiegelhalter *et al.*, 1993). This area is currently expanding from its current base in applied fields like expert systems to include some of the same positive aspects.

To see this, first note that the two communities are studying similar kinds of network. The graphical model given in Fig. 12, a Bayesian network, corresponds to the feed-forward network given by Ripley in Fig. 1. The nodes with double circles are deterministically computed from their inputs, as are nodes in a feed-forward network. The graphical model has a different final layer, however, to define the error model for the output variables y1, y2 and y3. In this case, the conditional probability for the output variables might be a softmax function of the hidden layer (Bridle, 1989, 1990). The arcs in the graphical model represent probabilistic rather than computational influence, although in the deterministic case they are equivalent. The shaded nodes represent observed variables (i.e. inputs), so this graphical model

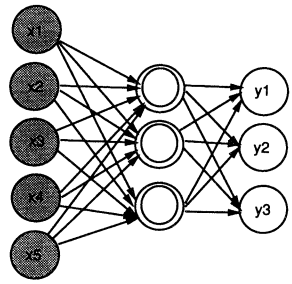

Fig. 12. Bayesian network modelling a feed-forward network

represents a conditional probability distribution equivalent to the feed-forward network. Similar correspondences hold for feed-forward networks for regression. Also, stochastic Hopfield networks correspond to undirected graphical models, which are a form of Markov random field. Of course, graphical models can model many other problems as well (Whittaker, 1990; Gilks *et al.*, 1993; Spiegelhalter *et al.*, 1993).

Next, you might have observed recent work on compiling data analysis algorithms from a specification given as a graphical model. For instance, BUGS by Spiegelhalter, Thomas and Gilks (Gilks *et al.*, 1993) generates a Gibbs sampler. Other more ambitious schemes under development by myself and others could easily compile algorithms like some neural network algorithms—and much more (Gilks *et al.*, 1993)—from simple or complex graphical specifications. For instance, gradients and Hessians of parameters in graphical models can be calculated in a similar fashion to feed-forward networks (Buntine and Weigend, 1994). Of course, these learning algorithms may also be computer intensive.

Chris Chatfield (University of Bath): I welcome this paper for focusing on neural networks (NNs), a topic of increasing importance which is still unfamiliar to many statisticians. The paper covers the application of NNs to classification problems but I would like to comment briefly on their proposed use in time series analysis and forecasting. Although tangential to the paper in some ways, this raises some general issues which are pertinent to the application of any technique to a new, unfamiliar, area.

Many people have tried to apply NNs to forecasting problems with mixed results (for example see Chatfield (1993)). A basic problem is that analysts with a computing background, who are experts on NNs, may know little about standard statistical methods, whereas statisticians generally know little about NNs at present. A secondary problem is that data come in all shapes and sizes and a technique which works well for one type of data may fail miserably with another. For example short seasonal time series, which customarily arise in sales forecasting, have quite different characteristics from long time series exhibiting non-linear properties. NNs generally give little, if any, improvement in forecasting accuracy for the former type of series (though they may give a better *fit* due to the large number of parameters available) but have been used successfully for the latter type of series (e.g. Weigend and Gershenfeld (1994)). In this sort of situation some researchers are liable to make exaggerated claims for some new technique, whereas other researchers may be inclined to resist claims for new methods in an unreasonable way. When the dust has settled, it is usually found that the new technique is neither a miraculous cure-all nor a complete disaster, but rather an addition to the analyst's toolkit which works well in some situations but not in others. Thus we need thorough comparative studies

(a) to assess the conditions under which a new technique should, and should not, be used and
(b) to make appropriate recommendations.

This paper and Ripley (1993) have made a good start but much work remains to be done.

Richard D. De Veaux (Princeton University), **Christian J. Darken** (Siemens Corporation, Princeton) **and Lyle H. Ungar** (University of Pennsylvania, Philadelphia): We would first like to congratulate the author on an impressive presentation of the problem of classification. As he himself mentioned that he may have missed or omitted several things in covering such a broad area, we feel compelled to comment on his brief treatment of radial basis functions (RBFs). The author seems to treat RBFs as being distinct from neural networks. This is misleading. Within the neural network community, which produced most

of the papers on RBFs that are cited, RBF networks are considered to be a type of feed-forward neural network of similar importance and with similar biological inspiration.

RBFs offer several advantages over sigmoidal networks. First, they are attractive for adaptive control, in part because fixing the basis functions makes the estimation of the coefficients a linear problem and, as such, amenable to the extensive analysis tools of adaptive control (Sanner and Slotine, 1992). Also, sigmoidal networks often create quite unreasonable divisions of the input spaces in the regions where no data are provided (see for example Kramer and Leonard (1990)), whereas RBFs put the basis functions close to the data and often give more sensible extrapolation. More importantly, unlike sigmoidal networks, they are easily adapted to estimate data density, thus providing a warning of extrapolation, and can be modified to give a local error estimate on each of their predictions (Leonard *et al.*, 1992). The lack of these features is a major fault of conventional sigmoidal networks.

The author objects to the emphasis on stochastic approximation in publications on neural networks and instead recommends general optimization techniques. Stochastic approximation uses an estimate of the gradient of the function to be optimized which is based on a single pattern, whereas conventional techniques evaluate the true gradient, which depends on the whole database of patterns. For very redundant databases, much of the effort spent by conventional techniques to measure the gradient accurately is wasted. Extremely large databases may require an impractically large time to evaluate the gradient, but stochastic approximation often performs very well on them (Moeller, 1992).

Finally, we would like to point out a problem of tree-based algorithms (e.g. classification and regression trees and multivariate adaptive and regression splines (MARS)) when the predictors are multicollinear. The fit of MARS can be significantly compromised by collinear predictors where neural nets perform well (De Veaux and Ungar, 1994). Although our study focused on prediction rather than classification, we suspect that the problem of multicollinearity may hold for classification applications as well.

Richard H. Glendinning (Defence Research Agency, Malvern): Model selection for neural networks poses formidable theoretical and practical problems. These stem from the following characteristics.

(a) The family of candidate models is typically large and non-nested. This makes a search of all models prohibitively expensive in many real world problems. One consequence is the widespread use of iterative techniques which perform a restricted search of candidate models. Recent contributions have emphasized the value of 'non-convergent' methods; see Finnoff *et al.* (1993). Here training is stopped when an estimate of predictive performance increases.

(b) The cardinality of the family of candidate models is typically large relative to the size of the training set. This may lead to overfitting and is analogous to the effect observed when several hypothesis tests are applied to the same data. This difficulty can be surmounted in practice by restricting the cardinality of the family of candidate models; see Hemerly and Davis (1991) for related ideas for linear models.

(c) The final characteristic is the inherent non-linearity of the problem and its nonparametric flavour. These characteristics have a profound effect on the form and properties of model selection criteria. For example, Cheng and Tong (1992) showed that cross-validation leads to a consistent estimate of model order for nonparametric time series. Related ideas are described in a neural network context in Liu (1993).

An understanding of the finite sample performance of model selection techniques in realistic scenarios is a relatively neglected area. Its importance is highlighted by Professor Ripley's remark on the non-existence of a globally optimal decision rule. This is relevant to model selection for linear regression (see Herzberg and Tsukanov (1986)) and autoregressive processes (see Glendinning (1993)).

Trevor Hastie (AT&T Bell Laboratories, Murray Hill) **and Robert Tibshirani** (University of Toronto): This paper provides considerable insight into the relationship between neural networks and statistical methods for high dimensional classification. We would like to comment on the optimal scoring and related approaches discussed in Section 5.

Breiman and Ihaka (1984) are appropriately credited for first pointing out that the link between optimal scoring and linear discriminant analysis can provide a non-linear version of discriminant analysis. Whereas their extensions relied on heuristics, Hastie, Tibshirani and Buja (1992) and Hastie, Buja and Tibshirani (1992) developed a general class of nonparametric discriminant models that included not only projection methods like multivariate adaptive regression splines (MARS) but also penalized regression. In this latter

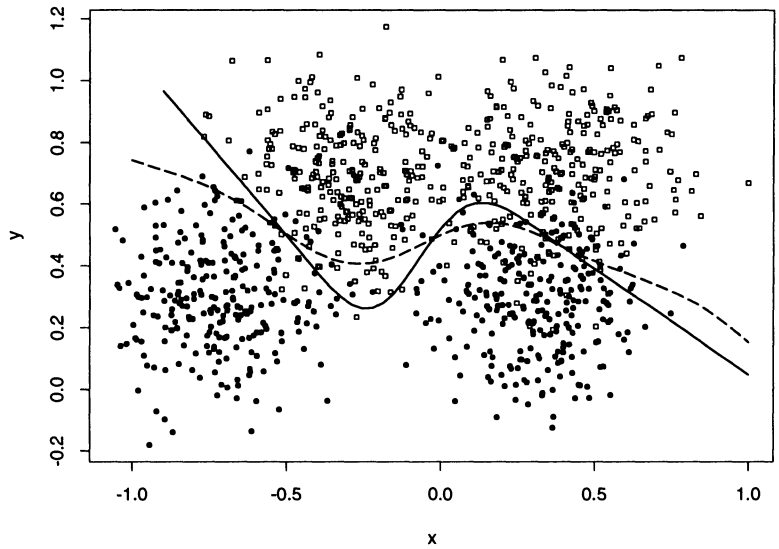

Fig. 13. Synthetic test data with the estimated Bayes decision boundary (———) and the FDA–BRUTO boundary (– – – –)

case, optimal scoring is not generally equivalent to a simple linear discriminant post-processing in the space of fitted values (as in the projection case of Section 5). Effectively this *flexible discriminant analysis* (FDA) involves expanding the input features into a large set of basis functions (usually splines and their tensor products), followed by a *penalized* discriminant analysis in the enlarged space. An important feature is that all this comes for free via optimal scoring—all that we need is the corresponding regression procedure that operates via basis expansion and penalized least squares.

Using additive spline regression, this procedure outperformed most of the procedures in Professor Ripley's Table 1, with a test error rate of 9.0%. Fig. 13 shows the fitted decision boundary, which is also among the most pleasing.

We find that this penalized version of optimal scoring is especially exciting for the classification of functions or images (Hastie, Buja and Tibshirani, 1992). In that problem the scenario is slightly different. We already have a rich (and highly correlated) feature set (neighbouring pixels or function values). Standard linear discriminant analysis (LDA) is too rough, and the coefficients can benefit from (spatial) regularization. The sonar example falls into this class. Even with standard ridge shrinking (from 60 parameters down to effectively 20), the performance of LDA improved from 25 errors down to 16.

An extension of optimal scoring to normal mixtures is described in Hastie and Tibshirani (1993).

Below Table 4, Professor Ripley suggests that the optimal scoring approach is generally inferior to log-linear models. Our limited experience does not support that view. FDA allows for dimension reduction and visualization, as in Fig. 9, a feature not shared by the multinomial model. Nor is the multinomial model necessarily better in terms of classification performance—see Table 5. Further work is needed to shed light on the relative merits of the FDA and multinomial approaches to non-linear classification.

G. J. McLachlan (University of Queensland): I congratulate Professor Ripley on a stimulating paper that provides a general framework within which neural networks are compared with other approaches to classification. This comparison will enhance and promote the use of non-linear classifiers whether they are obtained directly by neural network methods or by the other non-linear approaches surveyed. The author has compiled a comprehensive list of references in drawing together the vast literature on neural networks, discriminant analysis, pattern recognition and machine learning. Additional references on the use of regularization with non-linear classifiers may be found in chapter 5 of McLachlan (1992), including in particular the work of Friedman (1989) on regularization in the context of quadratic normal-theory-based discriminant analysis.

The paper contains many ideas of fundamental importance to non-linear classification. I shall confine my further comments to the problem of assessing the performance of a neural-network-based classifier

TABLE 5
Vowel recognition (Hastie, Tibshirani and Buja, 1992)†

Technique	*Error rates*	
	Training	*Test*
1, LDA	0.32	0.56
2, FDA–MARS (degree 2)	0.02	0.42
3, best reduced dimension (6) from row 2	0.13	0.39
4, Gaussian node network (528 hidden units)		0.45
5, multinomial using bases from row 2	0.02	0.54

†The data were obtained from a neural network bench-mark collection at Carnegie Mellon University (contributed by A. Robinson (1989) and maintained by S. Fahlman). In row 5, a multinomial model was fitted to the set of basis functions that were found by FDA using MARS (2). Row 4 is the best neural network classifier in the bench-mark collection.

in its application to future observations. The most commonly used method of estimating the actual error rate of a neural-network-based classifier appears to be the hold-out method, whereby the set of available classified data (i.e. the data of known origin) is split into disjoint training and test subsets. This method is inefficient in its use of the classified data, which are usually rare. This assessment problem in neural networks can be tackled by exploiting the approaches adopted by statisticians in estimating the actual error rates of a discriminant rule. In the latter context, methods such as cross-validation and the bootstrap have been used to provide useful error rate estimators, in particular to correct the apparent error rate for its optimistic bias when used as an estimator of the actual error rate. Given the ever-present danger of overfitting with neural networks, the bias correction of their apparent error rates as classifiers is of even greater importance. As noted by the author, in these non-linear and highly parameterized situations, bias correction of the apparent error rate via such methods as cross-validation is not straightforward. Even for standard discriminant analysis procedures, cross-validation provides an error rate estimator that is too variable in small samples. This has led to the development of bootstrap-based error rate estimators, including variants such as the 0.632 estimator of Efron (1983), which is almost the same as the linear combination (with weights 0.368 and 0.632) of the apparent error rate and the cross-validated error rate that leaves out half of the observations at a time. Similar linear combinations had been considered by McLachlan (1977) to correct the bias of the apparent error rate of (Fisher's) linear discriminant rule formed from homoscedastic normal training data. The suitability of estimators of this type is worth exploring for assessing the performances of neural-network-based classifiers.

Donald Michie (Turing Institute, Glasgow, and University of Edinburgh): Choice among classification methods has sometimes been contentious. Professor Ripley's authoritative and even-handed empiricism is therefore very timely.

Before leaving Strathclyde University for Oxford, Professor Ripley helped in efforts with European colleagues to launch the 'StatLog' project funded by the European Economic Community. Using data sets assembled in industry and medicine, the project made comparative trials of classification algorithms from machine learning, neural networks and statistics. StatLog sought a rational 'horses for courses' basis for choosing from today's algorithmic diversity. To the extent that this aim was achieved, StatLog's results complement Ripley's. Where they overlap his recent work, they are broadly confirmatory. The final report (Michie *et al.*, 1994) extends his coverage with additional algorithms and includes (albeit patchily) the case of unequal misclassification costs.

Ripley refers to what may be called 'mental fit':

> 'Only for predictive classification is a "black box" representation of f useful. (However, it is sometimes possible to approximate the black box in a more interpretable way, e.g. as a set of decision rules).'

In machine learning, mental fit is all-important. Industrial and medical clients quite commonly insist that classifiers be conceptually transparent, a consideration somewhat neglected in classical multivariate work. Ripley's mention of this consideration deserves emphasis. We need to establish more formally

those properties of algorithms that conduce to mental fit. His paper provides further inducement to the statistical and machine learning communities to familiarize themselves with each other's problems and methods.

Art B. Owen (Stanford University): I congratulate Professor Ripley for bringing the statistical and connectionist worlds closer together. My comments concern two areas touched on by Ripley: local optima in connectionist models and overfitting. My experience is with regression but I suspect that similar results hold for classification.

For regression, equation (1) has only one output y_k and ϕ_0 is the identity function. For redundant units with $w_{jk}=0$, the parameters 'inside ϕ_h' are not identifiable. This behaviour is similar to that seen in broken line regression (Davies, 1987; Hinkley, 1969; Knowles and Siegmund, 1989; Owen, 1991). Although $w_{jk}=0$ is unlikely, perhaps some $w_{jk}\doteq 0$ when many hidden units are employed. We might be suspicious of any hidden unit that reduces the sample squared error by less than a redundant unit might. Unfortunately, standard asymptotics do not say how much a redundant unit might reduce squared error.

Aldous's (1989) Poisson clumping heuristic provides one approach to this issue. Owen (1993a) applies that heuristic to find an expression for tail probabilities of the sum of squares explained by a redundant unit, when training one unit at a time. (Other units are linearized during such training.) The tail probability arises as the expected number of high local maxima of the sum of squares function, so it also gives some insight into the prevalence of local optima. Numerous spurious optima can defeat training by luring units away from true structure. Cross-validation exposes these decoys, but cannot recover the structure that they masked.

The clumping heuristic provides a diagnostic which shows where in the parameter space spurious units are likely to arise. For example, spuriously broken lines tend to bend near extreme input points, spuriously bent planes are more likely near the convex hull of the predictors and spurious sigmoidal units are more likely when their linear regions are in the centre of the input point cloud.

Owen (1993b) considers $d>1$ dimensional problems and finds that, for Gaussian inputs, the tail behaviour of sum of squares explained is like $A^d\chi^2_d$ for some A characterized by the method. Surprisingly some methods have A near 1. These tend to be 'heavy-tailed' methods, such as Cauchy sigmoids, hyperbolic approximations to broken planes and multiquadric radial basis functions. With 'short-tailed' methods such as piecewise linear sigmoids and ramps, and Gaussian radial basis functions, redundant units are much more likely to result in a large reduction in squared error and the squared error surface can have many more local extrema.

David H. Wolpert (Santa Fe Institute): I agree with Ripley's view that neural nets must be compared with other techniques (Wolpert, 1990) and that recent claims in the neural net literature that ML-II generically approximates hierarchical Bayes methods are overenthusiastic (Section 6.2; see Wolpert (1993a) and Wolpert *et al.* (1994)).

However, the claim of real world utility for Vapnik–Chervonenkis (VC) analysis (Section 7) managed to slip by Ripley's scrutiny. In the canonical version of this analysis, one assumes noise-free construction of the training set T. Therefore simple memorization suffices for feature vectors found in T, and the central object of interest is behaviour for feature vectors not in T. Accordingly, define c as the off-training set misclassification rate. Now examine the change in $E(c|\text{training set } T)$ when we use learning algorithm 1 rather than algorithm 2. In general, this change in risk will depend on the prior, as well as on the choice of the two learning algorithms. However, the change is zero when averaged over all prior distributions, *regardless of the two learning algorithms* (Wolpert, 1993b), i.e. there are 'just as many' priors for which algorithm 1 beats algorithm 2 as vice versa, for any pair of algorithms. (Similar results hold if only the size of T, p, is specified.)

In particular, say that we have an algorithm with a very small VC dimension which happens to have a very low apparent misclassification rate s on some large set T. We wish to compare this algorithm's extrapolation from T with that of some other arbitrary algorithm. Then there are as many priors for which the first algorithm's extrapolation is worse than the second algorithm's as there are for which the reverse is true. That s and the VC dimension are small gains us nothing.

How is this possible, given the VC theorem quoted by Ripley? The answer is that, when the theorem refers to 'probability', it does not explicitly state what that probability is conditioned on. If we exercise more care and rederive the VC theorem, we find that it does not refer to $P(c|T)$ or $P(c|p)$. Nor does it refer to a distribution like $P(c-s|s,p)$. Rather it and related results refer to distributions like $P(c-s|p)$, which are not conditioned on s.

The question for the practising statistician is which distribution tells you what you need to know? In particular, say that you are looking for a bound on a distribution; you want one which justifies the use of a learning algorithm in those cases where p is high but the algorithm results in low s. Given your goal, it is difficult to see why you should consider a distribution—like that in the VC theorem—which is not conditioned on low s.

The **author** replied briefly at the meeting, and later in writing as follows.

I would like to thank all the discussants for their contributions, especially for fleshing out the view of the neural networks field presented in the paper. The Society's page limit, although generous, does not suffice for such a large subject, and, as several discussants noted, my further thoughts have appeared elsewhere. In particular, Ripley (1994a, b, c, d) have been written in the year between the original submission of this paper and writing the reply, and Ripley (1994a) gives a complementary perspective written for the neural nets community on the experiments of Section 8. This reply also has a length limit, and for brevity I shall not comment on viewpoints outside the scope of my introduction, although I am grateful for their insights.

What is a neural network (artificial or digital if one prefers)?

Professor Whittle's viewpoint now seems old fashioned, although no less valid for that. It was certainly the hope of Frank Rosenblatt in the 1950s and has recurred in the field of machine learning. However,

(a) it is very difficult to analyse such general ideas, but restrictive analyses such as those of Minsky and Papert and of Judd have been universally discouraging and
(b) even given almost unlimited computational resources, such simple-minded algorithms do not seem to compare favourably with simple statistical analyses in practical examples.

(It is easy to invent a mouse-trap and to sell it to people who have never had one, difficult to succeed in a mature market!) I am not against stochastic algorithms *per se*; I just feel that almost always they are beaten comprehensively by even heuristic deterministic algorithms when people bother to compare them!

Radial basis functions

I do not consider radial basis functions (RBFs) to be (strict sense) neural networks as I see no convincing biological motivation for such functions in the original papers. (Their history goes back to the 'potential functions' of the Soviet school in the earlier 1960s. Even the 'locally tuned units' of Moody and Darken's analogy are special purpose and two-dimensional units, not models for general computation.) But there is a tendency for everything in a neural networks journal to be called a neural net, such as 'probabilistic neural nets' (Specht, 1990a, b) and 'regression neural networks' (Specht, 1991), which are just kernel methods. I see RBFs as another class of flexible approximators, with no computational advantage (since either optimization over centres or careful regularization with many centres is needed) and many more arbitrary choices (the metric, for example). But the emphasis here and in Ripley (1994a) is on how best to *use* such flexible approximators, not which to choose. The issue of extrapolation is taken up below, under fuller models.

Dr Tarassenko has made inspired use of the first stage of an RBF to learn about the class distributions over $\mathscr{X}$ in Roberts and Tarassenko (1993, 1994). I would myself prefer direct modelling by Gaussian mixtures in the sampling paradigm.

To answer Dr De Veaux and his colleagues it is always possible to use a series of samples of increasing size from the database while using conventional optimization methods (for both function values and gradient), and in my experience this significantly outperforms 'on-line' training. Let us compare the best of practice in each field.

Regularization

Regularization is important, and I prefer explicit methods to early stopping (the 'non-convergent' methods mentioned by Dr Glendinning). I was aware of the work quoted by Dr McLachlan, but I think that in his encyclopaedic book he misuses the term. Friedman's (1989) method is for normal class distributions, so barely 'non-linear', and the *shrinkage* used is within the sampling paradigm, and not to be confused with the regularization in function fitting discussed in Section 6.2. (Kimura *et al.* (1987)

had a very similar idea.) Such shrinkage methods have a long history; Campbell (1980) gives a practical example. But they do not seem anything like as powerful as the methods of this paper.

Assessment

As Professor Hand alludes, my statistical analysis here is sketchy, and in the verbal presentation I used data from Ripley (1994a) on the accuracy of the results, and I have taken the opportunity of adding these to the (new) table captions. That paper tries to explain a little experimental analysis to the neural network field. It also contains some work on using the estimates of the probabilities $p(C=k \mid \mathbf{X}=\mathbf{x})$ in assessment. The area appears promising, but so far I have failed to find a *simple* procedure which gives reliable indications for simple models with biased probability estimates, even on test sets. I am happier using such methods to assess the future performance of one classifier than to compare several. The methods mentioned by Dr McLachlan *are* in the text-book references at the start of Section 8, with practical examples.

Dr Taylor and Professor Michie refer to aspects of performance other than the error rate, and these can be important. Yet for many routine pattern recognition tasks the error rate and the 'reject rate' (the rate of declaring 'doubt') are crucial. Consider for example sorting letters and identifying fingerprints or deoxyribonucleic acid prints (Candela and Chellappa (1993) and Grother and Candela (1993), whose comparisons favour kernel methods). There small reductions in error rates translate into large cost savings.

Combination of classifiers

Mr Mayne asks how to combine different classifiers, and Professor Titterington mentions one view, that of Laveen Kanal. Rather less vague approaches are those of Wolpert (1992), Jacobs *et al.* (1991) and LeBlanc and Tibshirani (1993). The central insight of Dr Wolpert, taken up by LeBlanc and Tibshirani, is to use cross-validation or bootstrapping to obtain data on the mapping of the joint outputs of a set of classifiers to true class, and to model this by a very smooth classifier. For example, we might take a convex combination of the probabilities given by the classifiers, the combination being chosen by using regularization. In effect a second-layer classifier is trained to combine the first-layer classifiers. Jacobs *et al.* (1991) also combine simple classifiers (log-linear models) using a log-linear model, and Jordan and Jacobs (1993) add another two layers, *but* they train the whole combined classifier simultaneously.

Professor Brown mentions Bayesian model choice. I have become convinced that strict Bayesians do not choose models; they average posterior probabilities over models as his equation shows. (This seems unappreciated by most 'Bayesians' in the neural networks and signal processing fields.) This is clearly the best form of combination of classifiers, but $P(\text{model} \mid \text{data})$ can involve intractable integrations, so the stacked generalization approach can be seen as a way to estimate those probabilities.

Evolution of classes

Evolution of classes (another of Mr Mayne's questions) can only be tackled within the sampling paradigm, since there it is possible to update the parameters of the class distributions from unclassified future samples. Two practical examples are symbol recognition (Hjort, 1986; Eikvil *et al.*, 1992) where styles change slowly over times, and in classifying magnetic resonance imaging brain slices (Storvik *et al.*, 1992) where the effect of each 'exposure' has to be taken into account by retraining for that 'exposure'.

Predictive Bayes

The predictive Bayes approach is appealing (Titterington, Brown), and I have recently achieved some success by approximating the posterior distribution of the weights by a mixture of Gaussian distributions centred on local minima of the (regularized) fit criterion, i.e. on peaks of the posterior density (Ripley, 1994a). As Professor Breiman guesses, local minima *are* a problem, much ignored. I do my optimization carefully, including checking if I have reached a local minimum by checking the Hessian, and using many (often hundreds of) starting points. (There are at least 22 distinct local minima in the forensic science example, and rarely do I find just one.) These local minima do not all give equally good classifiers (Table 3), and some are at the top of *much* narrower peaks of the posterior density than others so correspond to negligible components of my Gaussian mixture approximations. For the forensic science example, I obtained error rates of (28%, 12%) for Table 4 for the predictive approach for a neural network with six hidden units.

My comments in Section 6.2 on the work of MacKay, Buntine and Weigend tell only half the story. I do find the approximation by a single λ poor, but MacKay also advocates multiple-λ approximations. In the Buntine and Weigend (1991) approach one can indeed integrate over the non-informative hyperprior analytically, but then it seems very much more difficult to approximate the posterior distribution of

the weights by components around peaks (and the mode alone seems inappropriate as a single set of weights). The approach of the preceding paragraph allows an approximate integration over an informative prior at a little extra computational cost (just average the mixtures of Gaussian distributions over several samples of λ).

Fuller models

I have great sympathy with the modelling approach sketched by Dr Buntine and Dr Wilks, and expounded brilliantly by Spiegelhalter *et al.* (1993), and indeed I too have tried to place neural networks within that context in Ripley (1994c). It includes joint modelling of $\mathbf{X}$ and Y, which is a considerable advantage in considering *outliers* in $\mathscr{X}$, i.e. extrapolation. In the diagnostic paradigm the predictions from outlying $\mathbf{x}$ can be extreme. This is one claimed advantage of the 'local' RBF approach, which if used for log-probability modelling will revert to the prior under extrapolation. This is not necessarily what is required, as for example with a one-dimensional $\mathbf{x}$ it will force non-monotonic behaviour. Only a fuller model can include one's intuition here. However, to model well does demand prior knowledge of structure, and it is not clear to me that in the vast majority of examples (including almost all that I have tried) we have that knowledge. It may be possible to induce the *structure* of a Bayes net from data (Cooper and Herskovits, 1992; Pearl, 1988; Spirtes *et al.*, 1993) but this may need much more data than we typically have (let alone computer time).

Model selection

I know of no useful results on the trade-offs between many nodes in one layer *versus* many layers (Critchley). What experience does show is that training multilayer networks is very difficult, unless they are of a very heavily constrained form. (I suspect that even finding deep local minima has never been achieved.) Of course, the predictive Bayesian paradigm can finesse model choice (above).

Generalization

The generalization results of Section 7 *are* frequentist, over training sets $\mathscr{T}$, and there is a *usually* in the critical sentence to reflect this—we will never know if this particular $\mathscr{T}$ is 'typical'. At the time of writing the paper I had not met the abuses alluded to by Dr Wolpert, but I have since. Holden and Anthony (1993) specifically claim that *if* the network can be trained for low apparent error rate, then the results bound the true error rate. This conditional statement is not a valid deduction from the theory and depends on the probability of trainability being bounded below to be correctable.

Later results on VC-like bounds appear in Haussler (1992) within a general decision theory framework, and the mathematical text of Anthony and Biggs (1993) covers the simpler results of the field. These accounts are not as cavalier as the work alluded to by Dr Wolpert, but in any case these results are not practical unless VC-like quantities are available for practical networks (such as feed-forward networks with sigmoids, RBFs with variable centres and regularized networks).

Optimal scoring

I agree with Professor Breiman and Dr Hastie and Dr Tibshirani that we need more experience with optimal scoring and log-probability methods. But the latter seriously misquote me, and until we have fully operational methods for multivariate adaptive regression splines and projection pursuit regression of both types the verdict remains unclear. However, only the log-probability approach readily allows the use of (approximate) predictive Bayes methods.

Data availability

All the data sets were available to discussants and will remain available for the foreseeable future. They are available from the Internet machine address markov.stats.ox.ac.uk as a compressed shar file pub/neural/RSS.data.Z. Although you should not use this information, the synthetic data set was generated from equal mixtures of normal distributions with centres $(-0.7, 0.3)$ and $(0.3, 0.3)$ in class 0 and $(-0.3, 0.7)$ and $(0.4, 0.7)$ in class 1, with variances 0.03.

REFERENCES IN THE DISCUSSION

Aitchison, J. and Kay, J. W. (1975) Principles, practice and performance in decision making in clinical medicine. In *The Role and Effectiveness of Theories of Decision in Practice* (eds D. J. White and K. C. Bowen), pp. 252–272. London: Hodder and Stoughton.

Aldous, D. (1989) *Probability Approximations via the Poisson Clumping Heuristic*. New York: Springer.

Amari, S. (1985) *Differential Geometrical Methods in Statistics*. New York: Springer.

——— (1993) Mathematical methods of neurocomputing. In *Networks and Chaos—Statistical and Probabilistic Aspects* (eds O. E. Barndorff-Nielsen, J. L. Jensen and W. S. Kendall), pp. 1–39. London: Chapman and Hall.

Anthony, M. and Biggs, N. L. (1992) *Computational Learning Theory: an Introduction*. Cambridge: Cambridge University Press.
Becker, S. (1991) Unsupervised learning procedures for neural networks. *Int. J. Neural Syst.*, **2**, 17–33.
Becker, S. and Hinton, G. E. (1992) Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, **355**, 161–163.
Bienenstock, E. L., Cooper, L. N. and Munro, P. W. (1982) Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neursci.*, **2**, 32–48.
Breiman, L. (1994) Discussion on Neural networks: a review from a statistical perspective (by B. Cheng and D. M. Titterington). *Statist. Sci.*, **9**, in the press.
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 *Neuro-computing: Algorithms, Architectures and Applications* (eds F. Fogleman-Soulie and J. Hérault). New York: Springer.
———(1990) Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Advances in Neural Information Processing Systems 2* (ed. D. S. Touretzky), pp. 211–217. San Mateo: Morgan Kaufmann.
Brown, P. J. (1982) Multivariate calibration (with discussion). *J. R. Statist. Soc.* B, **44**, 287–321.
———(1993) *Measurement, Regression, and Calibration*. Oxford: Clarendon.
Buntine, W. L. and Weigend, A. S. (1991) Bayesian back-propagation. *Complex Syst.*, **5**, 603–643.
———(1994) Calculating second derivatives on feed-forward networks. *IEEE Trans. Neural Netwks*, to be published.
Campbell, N. A. (1980) Shrunken estimators in discriminant and canonical variate analysis. *Appl. Statist.*, **29**, 5–14.
Candela, G. T. and Chellappa, R. (1993) Comparative performance of classification methods for fingerprints. *Report NISTIR 5163*. US National Institute of Standards and Technology, Gaithersburg.
Chatfield, C. (1993) Neural networks: forecasting breakthrough or passing fad? *Int. J. Forecast.*, **9**, 1–3.
Cheng, B. and Titterington, D. M. (1994) Neural networks: a review from a statistical perspective (with discussion). *Statist. Sci.*, **9**, in the press.
Cheng, B. and Tong, H. (1992) On consistent nonparametric order determination and chaos. *J. R. Statist. Soc.* B, **54**, 427–449.
Cook, R. D. (1993a) On the interpretation of regression plots. *J. Am. Statist. Ass.*, to be published.
———(1993b) Graphics for studying net effects of regression predictors (with discussion). *Statist. Sin.*, to be published.
Cook, R. D. and Wetzel, N. (1993) Exploring regression structure with graphics (with discussion). *Test*, **2**, no. 1, in the press.
Cooper, G. F. and Herskovits, E. (1992) A Bayesian method for the induction of probabilistic networks from data. *Mach. Learn.*, **9**, 309–347.
Critchley, F., Marriott, P. K. and Salmon, M. H. (1993) An elementary account of Amari's expected geometry. *Statistics Research Report*. University of Birmingham, Birmingham.
Davies, R. B. (1987) Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika*, **74**, 247–254.
De Veaux, R. D. and Ungar, L. H. (1994) Multicollinearity: a tale of two nonparametric regressions. In *Selecting Models from Data: AI and Statistics IV* (eds P. Cheeseman and R. W. Oldford), pp. 293–302. New York: Springer.
Efron, B. (1983) Estimating the error rate of a prediction rule: improvement on cross-validation. *J. Am. Statist. Ass.*, **78**, 316–331.
Eikvil, L., Holden, M. and Storvik, G. (1992) Methods for updating of model parameters applied within the area of symbol recognition. *Report 859*. Norwegian Computing Center, Oslo.
Evans, D. G., Scotter, C. N. G., Day, L. Z. and Hall, M. N. (1993) Determination of the authenticity of orange juice by discriminant analysis of near infrared spectra: a study of pretreatment and transformation of spectral data. *J. Near Infrared Spectrosc.*, **1**, 33–44.
Finnoff, W., Hergert, F. and Zimmermann, H. G. (1993) Improving model selection by nonconvergent methods. *Neural Netwks*, **6**, 771–783.
Foldiak, P. (1990) Forming sparse representations by local anti-hebbian learning. *Biol. Cyb.*, **64**, 165–170.
Friedman, J. H. (1989) Regularized discriminant analysis. *J. Am. Statist. Ass.*, **84**, 165–175.
Galland, C. C. and Hinton, G. E. (1989) Discovering high order features with mean field modules. *Technical Report*. Connectionist Research Group, University of Toronto, Toronto.
Geman, S., Bienenstock, E. and Doursat, R. (1992) Neural networks and the bias/variance dilemma. *Neural Computn*, **4**, 1–58.
Gilks, W. R., Clayton, D. G., Spiegelhalter, D. J., Best, N. G., McNeil, A. J., Sharples, L. D. and Kirby, A. J. (1993) Modelling complexity: applications of Gibbs sampling in medicine. *J. R. Statist. Soc.* B, **55**, 39–52.
Gilks, W. R., Oldfield, L. and Rutherford, A. (1989) Statistical analysis. In: *Leucocyte Typing IV* (eds W. Knapp, B. Dörken, W. R. Gilks, E. P. Rieber, R. E. Schmidt, H. Stein and A. E. G. Kr. von dem Borne), pp. 6–12. Oxford: Oxford University Press.
Glendinning, R. H. (1993) Model selection for infinite variance time series. To be published.

Grother, P. J. and Candela, G. T. (1993) Comparison of handprinted digit classifiers. *Report NISTIR 5209*. US National Institute of Standards and Technology, Gaithersburg.
Haralick, R. M. (1979) Statistical and structural approaches to texture. *Proc. IEEE*, **67**, 786–804.
Hastie, T., Buja, A. and Tibshirani, R. (1992) Penalized discriminant analysis. To be published.
Hastie, T. and Tibshirani, R. (1993) Discriminant analysis by mixture estimation. To be published.
Hastie, T., Tibshirani, R. and Buja, A. (1992) Flexible discriminant analysis by optimal scoring. To be published.
Haussler, D. (1992) Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inform. Computn*, **100**, 78–150.
Hemerly, E. M. and Davis, M. H. A. (1991) Recursive order estimation of autoregressions without bounding the model set. *J. R. Statist. Soc.* B, **53**, 201–210.
Herzberg, A. M. and Tsukanov, A. V. (1986) A note on modifications of the jackknife criterion for model selection. *Util. Math.*, **29**, 209–216.
Hinkley, D. V. (1969) Inference about the intersection in two phase regression. *Biometrika*, **56**, 495–504.
Hjort, N. L. (1986) Notes on the theory of statistical symbol recognition. *Report 778*. Norwegian Computing Center, Oslo.
Holden, S. B. and Anthony, A. (1993) Quantifying generalization in linearly weighted neural networks. *Technical Report CUED/F-INFENG/TR.113*. Department of Engineering, University of Cambridge, Cambridge.
Hornick, K. and Kuan, C. M. (1992) Convergence analysis of local feature extraction algorithms. *Neural Netwks*, **5**, 229–240.
Intrator, N. (1990) Feature extraction using unsupervised learning. *Neural Computn*, **4**, 98–107.
Intrator, N. and Cooper, L. N. (1992) Objective function formulation of the BCM theory of visual cortical plasticity: statistical connections and stability conditions. *Neural Netwks*, **5**, 3–17.
Intrator, N. and Gold, J. I. (1993) Three dimensional object recognition of gray level images: the usefulness of distinguishing features. *Neural Computn*, **5**, 61–74.
Jacobs, R. A., Jordan, M. I., Nowlan, S. J. and Hinton, G. E. (1991) Adaptive mixtures of local experts. *Neural Computn*, **3**, 79–87.
Jordan, M. I. and Jacobs, R. A. (1993) Hierarchical mixtures of experts and the EM algorithm. Submitted to *Neural Computn*.
Kanal, L. (1993) On patterns, categories and alternate realities. *Pattn Recogn Lett.*, **14**, 241–255.
Kimura, F., Takashina, K., Tsuruoka, S. and Miyake, Y. (1987) Modified quadratic discriminant functions and the application to Chinese character recognition. *IEEE Trans. Pattn Anal. Mach. Intell.*, **9**, 149–153.
Knowles, M. and Siegmund, D. O. (1989) On Hotelling's approach to testing for a nonlinear parameter in regression. *Int. Statist. Rev.*, **57**, 205–220.
Kramer, M. A. and Leonard, J. A. (1990) Diagnosis using backpropagation neural networks: analysis and criticism. *Comput. Chem. Engng*, **14**, 1323–1338.
LeBlanc, M. and Tibshirani, R. (1993) Combining estimates in regression and classification. *Technical Report*. Department of Statistics, University of Toronto, Toronto.
Leonard, J. A., Kramer, M. A. and Ungar, L. H. (1992) Using radial basis functions to approximate a function and its error bounds. *IEEE Trans. Neural Netwks*, **3**, 624–627.
Lindley, D. V. (1956) On a measure of the information provided by an experiment. *Ann. Math. Statist.*, **27**, 986–1005.
Linsker, R. (1988) Self-organization in a perceptual network. *Computer*, **21**, 105–117.
———(1992) Local synaptic learning rules suffice to maximize mutual information in a linear network. *Neural Computn*, **4**, 691–702.
Lippmann, R. P. (1989) Pattern classification using neural networks. *IEEE Communs Mag.*, **27**, 47–64.
Liu, Y. (1993) Neural network model selection using asymptotic jackknife estimator and cross-validation method. In *Advances in Neural Information Processing Systems 5* (eds S. J. Hanson, J. D. Cowan and C. L. Giles), pp. 599–606.
Luttrell, S. P. (1989) Hierarchical vector quantisation. *Proc. IEEE*, part I, **136**, 405–413.
McLachlan, G. J. (1977) A note on the choice of a weighting function to give an efficient method for estimating the probability of misclassification. *Pattn Recogn*, **9**, 147–149.
———(1992) *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley.
Michie, D., Spiegelhalter, D. S. and Taylor C. C. (eds) (1994) *Machine Learning, Neural and Statistical Classification*. Chichester: Horwood. To be published.
Miller, W. T., Glanz, F. H. and Kraft, L. G. (1987) Application of a general learning algorithm to the control of robotic manipulators. *Int. J. Robot. Res.*, **6**, 84–98.
Moeller, M. (1992) Supervised learning on large redundant training sets. In *Neural Networks for Signal Processing II* (eds S. Y. Kung, F. Fallside, J. A. Sorenson and C. A. Kamm). Piscataway: Institute of Electrical and Electronics Engineers.
Oja, E. (1989) Neural networks, principal components and subspaces. *Int. J. Neural Syst.*, **1**, 61–68.
Owen, A. (1991) Comment on Multivariate adaptive regression splines (by J. H. Friedman). *Ann. Statist.*, **19**, 102–112.
———(1993a) Poisson clumping and redundant units. *Technical Report 427*. Department of Statistics, Stanford University, Stanford.

———(1993b) Redundant units in high dimensions. *Technical Report 432*. Department of Statistics, Stanford University, Stanford.
Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo: Morgan Kaufmann.
Ripley, B. D. (1993) Statistical aspects of neural networks. In *Networks and Chaos—Statistical and Probabilistic Aspects* (eds O. E. Barndorff-Nielsen, J. L. Jensen and W. S. Kendall), pp. 40–123. London: Chapman and Hall.
———(1994a) Flexible non-linear approaches to classification. In *From Statistics to Neural Networks* (eds V. Cherkassky, J. H. Friedman and H. Wechsler). New York: Springer.
———(1994b) Network methods in statistics. In *Probability, Statistics, Optimization: a Tribute to Peter Whittle* (ed. F. P. Kelly). Chichester: Wiley.
———(1994c) Discussion on Neural networks: a review from a statistical perspective (by B. Cheng and D. M. Titterington). *Statist. Sci.*, **9**, in the press.
———(1994d) Choosing network complexity. In *Adaptive Computing and Information Processing* (eds J. G. Taylor, A. Gammerman and V. Rayward-Smith). Uxbridge: Unicom.
Roberts, S. and Tarassenko, L. (1993) Automated sleep EEG analysis using an RBF network. In *Neural Network Applications* (ed. A. F. Murray). Boston: Kluwer.
———(1994) A probabilistic resource allocating network for novelty detection. *Neural Computn*, to be published.
Robinson, A. (1989) Dynamic error propagation networks. *PhD Thesis*. Department of Electrical Engineering, University of Cambridge, Cambridge.
Rumelhart, D. E. and Zipser, D. (1985) Feature discovery by competitive learning. *Cogn. Sci.*, **9**, 75–112.
Sanger, T. D. (1989) Optimal unsupervised learning in a single-layer linear feed-forward neural network. *Neural Netwks*, **2**, 459–473.
Sanner, R. M. and Slotine, J.-J. E. (1992) Gaussian networks for direct adaptive control. *IEEE Trans. Neural Netwks*, **3**, 837–863.
Schlossman, S., Boumsell, L., Gilks, W. R., Harlan, J., Kishimoto, T., Morimoto, C., Ritz, J., Shaw, S., Silverstein, R., Springer, T., Tedder, T. and Todd, R. (eds) (1994) *Leukocyte Typing V*. Oxford: Oxford University Press.
Smith, A. F. M. and Spiegelhalter, D. J. (1981) Bayesian approaches to multivariate structure. In *Interpreting Multivariate Data* (ed. V. Barnett), pp. 335–348. Chichester: Wiley.
Specht, D. F. (1990a) Probabilistic neural networks. *Neural Netwks*, **3**, 109–118.
———(1990b) Probabilistic neural networks and the polynomial Adaline as complementary techniques for classification. *IEEE Trans. Neural Netwks*, **1**, 111–121.
———(1991) A general regression neural network. *IEEE Trans. Neural Netwks*, **2**, 568–576.
Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L. and Cowell, R. G. (1993) Bayesian analysis in expert systems. *Statist. Sci.*, **8**, 219–283.
Spirtes, P., Glymour, C. and Scheines, R. (1993) Causality, prediction and search. *Lect. Notes Statist.*, **81**.
Storvik, G., Holden, M. and Bosnes, V. (1992) Improving statistical image classification by updating model parameters using unclassified pixels. *Report 857*. Norwegian Computing Center, Oslo.
Titterington, D. M., Murray, G. D., Murray, L. S., Spiegelhalter, D. J., Skene, A. M., Habbema, J. D. F. and Gelpke, G. J. (1981) Comparison of discrimination techniques applied to a complex data set of head injured patients (with discussion). *J. R. Statist. Soc.* A, **144**, 145–174.
Unser, M. (1986) Sum and difference histograms for texture classification. *IEEE Trans. Pattn Anal. Mach. Intell.*, **8**, 118–125.
Wahba, G., Gu, C., Wang, Y. and Chappell, R. (1993) Soft classification, a.k.a. penalized log likelihood and smoothing spline analysis of variance. In *Proc. Wkshp Supervised Machine Learning* (eds D. Wolpert and A. Lapedes). Reading: Addison-Wesley.
Weigend, A. S. and Gershenfeld, N. A. (eds) (1994) *Time Series Prediction: Forecasting the Future and Understanding the Past*. Reading: Addison-Wesley.
Weszka, J. S., Dyer, C. R. and Rosenfeld, A. (1976) A comparative study of texture measures for terrain classification. *IEEE Trans. Syst. Man Cyb.*, **6**, 269–285.
Whittaker, J. (1990) *Graphical Models in Applied Multivariate Statistics*. Chichester: Wiley.
Wolpert, D. H. (1990) Constructing a generalizers superior to NETtalk via a mathematical theory of generalization. *Neural Netwks*, **3**, 445–452.
———(1992) Stacked generalization. *Neural Netwks*, **5**, 241–259.
———(1993a) On the use of evidence in neural networks. In *Advances in Neural Information Processing Systems 5* (eds S. Hanson *et al.*), pp. 539–546.
———(1993b) On overfitting avoidance as bias. *Technical Report 93-03-016*. Santa Fe Institute, Santa Fe.
Wolpert, D. H., Strauss, C. E. and Wolf, D. R. (1994) What Bayes has to say about the evidence procedure. In *Maximum Entropy and Bayesian Methods* (ed. G. Heidbreder). To be published.
Xu, L. and Yuille, A. (1992) Robust PCA learning rules based on a statistical physics approach. In *Proc. IJCNN Int. Joint Conf. Neural Networks, Baltimore*, book I, pp. 812–817. New York: Institute of Electrical and Electronics Engineers.