Flexible Non-linear Approaches to Classification

B. D. Ripley

Professor of Applied Statistics, University of Oxford
1 South Parks Road, Oxford OX1 3TG, UK. ripley@stats.ox.ac.uk

To appear in

From Statistics to Neural Networks. Theory and Pattern Recognition Applications. eds V. Cherkassky, J. H. Friedman and H. Wechsler. ASI Proceedings, subseries F, Computer and Systems Sciences. Springer-Verlag.

The data are currently available by anonymous ftp in directory
`markov.stats.ox.ac.uk:pub/neural/ASI`
See its `README`.

Flexible Non-linear Approaches to Classification

B. D. Ripley

Professor of Applied Statistics, University of Oxford
1 South Parks Road, Oxford OX1 3TG, UK. ripley@stats.ox.ac.uk

Abstract. Neural networks are one of a class of classifiers which construct a non-linear function from inputs to targets. There are a series of questions common to all members of the class, including how best to use the outputs to classify, how to fit the class of functions and also how to choose between classes. These questions are explored via some theory and three real examples. We also discuss the analysis of comparative experiments.

1 Statistical pattern recognition

The most widely accepted paradigm in classification is that of statistical decision theory (Berger, 1985; McLachlan, 1992; Ripley & Hjort, 1994) also known as statistical pattern recognition (Fukunaga, 1990). We suppose that there is a feature space $\mathcal{X}$ of potential observations, and to each we should assign a label, either that of one of K classes or 'doubt' $\mathcal{D}$ or 'outlier' $\mathcal{O}$. 'Doubt' is used when more than one class seems plausible and 'outlier' when no class is plausible. (Doubt options are particularly useful when there is a second line of classification, for example to make a more expensive test or to consult a human expert.) Let $\mathcal{Y} = \{1, \dots, K, \mathcal{D}, \mathcal{O}\}$ be the space of possible decisions; a *classifier* is then a map from $\mathcal{X}$ to $\mathcal{Y}$.

This paradigm assumes cases $(\mathbf{X}, Y)$ are drawn independently from a probability distribution; we call the random variable $\mathbf{X}$ the *pattern* and Y the true decision (which usually does not take the value $\mathcal{D}$). The *training set* $\mathcal{T}$ is a set of n classified cases. Future cases will present a pattern $\mathbf{X}$ for the classifier to compute a decision.

The probability distribution of pairs $(\mathbf{X}, Y)$ can be specified in a number of ways. The *sampling paradigm* (in the terminology of Dawid, 1976) gives a prior distribution π_k over classes and the class-conditional densities $p_k(\mathbf{x})$. The *diagnostic paradigm* gives the posterior densities $p(k \mid \mathbf{x})$ and the marginal density $p(\mathbf{x})$ of $\mathbf{X}$. These specify the same information, for the joint density is $\pi_k p_k(\mathbf{x}) = p(k \mid \mathbf{x}) p(\mathbf{x})$.

Statistical decision theory assigns costs to decisions. We assume that correct decisions have no cost, incorrect ones unit cost and 'doubt' costs $0 < d \le 1$. Then the rule with minimum expected cost, the *Bayes rule*, is specified by

$$y(\mathbf{x}) = \begin{cases} k & \text{if } p(k \mid \mathbf{x}) = \max_j p(j \mid \mathbf{x}) > 1 - d \\ \mathcal{D} & \text{if } \max_j p(j \mid \mathbf{x}) \le 1 - d \end{cases}$$

Note that the Bayes rule does not depend (directly) on $p(\mathbf{x})$. The expected loss under the Bayes rule, the *Bayes risk* represents the best achievable performance in the problem. (We can include different costs for incorrect decisions by replacing $p(k \mid \mathbf{x})$ by $1 - \sum_{k \neq j} C_{jk} p(j \mid \mathbf{x})$, where C_{kj} is the cost of confusing class k with j.) The posterior probabilities $p(k \mid \mathbf{x})$ are central.

It is less clear-cut how to treat outliers in the decision-theory framework. The purist approach would be to consider outliers to form another class, but then it is necessary to specify the class-conditional density $p_{\mathcal{O}}(\mathbf{x})$ of outliers. A less rigorous approach is to class as outliers those patterns $\mathbf{x}$ for which $p(\mathbf{x})$ is small. This needs the overall density $p(\mathbf{x})$ to be estimated, for example by a kernel density estimator. (Smyth, 1993, takes this approach.)

Most classifiers can be seen as approximations to the Bayes rule. Let $f_k(\mathbf{x}) = p(k \mid \mathbf{x}) = E(I(Y = k) \mid \mathbf{X} = \mathbf{x})$ and $\mathbf{f}(\mathbf{x}) = (f_k(\mathbf{x}))$. The Bayes rule chooses a maximizer of $f_k(\mathbf{x})$ over k. So if we can estimate $\mathbf{f}(\mathbf{x})$ or any monotonic function of it such as $g_k(\mathbf{x}) = -\log f_k(\mathbf{x})$, we can find an approximation to the Bayes rule. Define *target* $\mathbf{t}_k$ to be the kth unit vector. Since

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{t}_k\|^2 = -2 f_k(\mathbf{x}) + 1 + \|\mathbf{f}(\mathbf{x})\|^2$$

the Bayes' rule amounts to finding the nearest target to $\mathbf{f}(\mathbf{x})$. (Note that unlike common practice for neural networks, doubt is not expressed by the distance from a target.)

The difficulties emerge when (as is usual) the distributions are unknown but have to be estimated from the training set $\mathcal{T}$. Suppose that we have a family of distributions parametrized by θ. The commonest approach is the *plug-in* approach, to compute the Bayes rule based on $p(k \mid \mathbf{x}; \hat\theta)$, where $\hat\theta$ is an estimate of θ. This is not strictly correct as the uncertainty in $\hat\theta$ is not taken into account. The *predictive* approach uses

$$\begin{aligned} P(Y = k \mid \mathbf{X} = \mathbf{x}, \mathcal{T}) &= \int p(k \mid \mathbf{x}; \theta) p(\theta \mid \mathcal{T})\, d\theta \\ &\propto \int p(k \mid \mathbf{x}; \theta \prod_{p=1}^{n} p(\mathbf{x}^p, y^p; \theta))\ p(\theta)\, d\theta \\ &\propto \int p(k \mid \mathbf{x}; \theta) \prod_{p=1}^{n} p(y^p \mid \mathbf{x}^p; \theta) p(\mathbf{x}^p; \theta)\ p(\theta)\, d\theta \qquad (1) \end{aligned}$$

and this needs a prior distribution over θ. This full Bayesian approach has been tried for regression neural networks (e.g. Neal, 1993, who uses a Monte Carlo method.). MacKay (1992a,b,c) uses a series of approximations to this approach to make it more computationally tractable. Note that 'model choice' such as choosing the number of hidden units or the network topology is contained in θ; in principle we do not select models but take a weighted average over all models. (In practice we average over a few good models.)

In the diagnostic paradigm it is usual to work conditionally on $\mathbf{X}$. As we have seen, the Bayes rule does not depend on the marginal distribution of $\mathbf{X}$, but the

likelihood of θ given $\mathcal{T}$ will unless $p(\mathbf{x}; \theta)$ does not depend on θ, and so there is usually a loss of information (in the technical sense), balanced by having to specify less of the model. One loss which may be considerable is the inability to estimate $p(\mathbf{x}; \theta)$ from unclassified samples. (In some fields examples are readily available but producing accurate classifications is slow, expensive or both.) We have already noted that $p(\mathbf{x})$ may be needed to detect outliers.

There are pure nonparametric approaches based on the sampling paradigm estimating the class-conditional densities, including kernel discrimination (Hand, 1982; Coomans & Broeckaert, 1986, also known as *probabilistic neural networks*, Sprecht, 1990; Wasserman, 1993) and using mixture distributions (Titterington *et al.*, 1985). The latter is closely related to, but differs from, the use of Gaussian radial basis functions.

Since the Bayes risk represents the best possible performance, it is worth making some effort to estimate it. The theory of k-nearest neighbour rules provides asymptotic bounds which with careful choice of metric can give a useful idea of the Bayes risk (Fukunaga, 1990, §7.4).

Discrimination

In some problems the primary aim of producing a classifier is not to predict future observations but to encapsulate the essential differences between the groups. In the statistical literature this is sometimes known as *discrimination* against *allocation* or *classification*, but the concepts and ideas are often jumbled.

It seems not to be widely known that Fisher's original purpose in studying the (in)famous Iris data in his 1936 paper was discrimination rather than classification. His genetical hypothesis was that *Iris versicolor* had been formed as a 2:1 mixture of *I. virginica* and *I. setosa*. He used the 'Fisher linear discriminant' to find the best linear combination of measurements separating the two putative parents, and then test whether *I. versicolor* lay 2:1 between them on that combination. (It did.)

2 Flexible function fitting

Several families of functions $\mathbf{f}(\ ; \theta) : \mathcal{X} \to \mathbb{R}^d$ are known to be universal approximators over compact sets, including single-hidden-layer neural networks (e.g. Cybenko, 1989; Jones, 1990), projection-pursuit regression (Friedman & Stuetzle, 1981; Diaconis & Shahshahani, 1984) and sub-classes of radial basis functions (Girosi & Poggio, 1990; Park & Sandberg, 1991, 1993). There are other recent approaches such as MARS (Friedman, 1991) and the Π and hinging hyperplane methods (Breiman, 1991, 1993). Splines (Wahba, 1990) also have a place although they are much less convenient in more than one dimension. The close conceptual relationship between projection-pursuit regression and neural networks is discussed by Hwang *et al.* (1992, 1993) and Ripley (1993a, 1994b).

There are several papers (e.g. Gish, 1990; Hampshire & Pearlmutter, 1990; Miller *et al.*, 1993; Richard & Lippmann, 1991; Ruck *et al.*, 1990) which consider large samples and show that provided the parametric family is such that $f(\mathbf{x};\theta)$ can approximate the true $f(\mathbf{x})$, the classifier will approximate the Bayes rule. Of course, in large samples we expect the distribution of $p(\theta \mid \mathcal{T})$ to concentrate near a single point, and subject to continuity conditions the plug-in and predictive approaches become equivalent.

For finite samples there are trade-offs between the flexibility of $\mathbf{f}(\mathbf{x};\theta)$ and the spread of $p(\theta \mid \mathcal{T})$ (e.g. Geman *et al.*, 1992). We will need to impose limits on the degree of flexibility of the function, such as restrictions on the number of parameters (for example by restricting the number of hidden units or the number of radial bases) or to penalize rough functions. For the latter we can impose regularization penalties (Bishop, 1991, 1993; Poggio & Girosi, 1990) or use a prior over θ (Buntine & Weigend, 1991; MacKay, 1992a,b). An example is weight decay for neural networks which I see as important enough to use routinely (Ripley, 1993a,b). Regularization penalties themselves usually have one or more free parameters, which can be seen as parameters in a prior and themselves gives a (hyper)prior.

We can learn a lot from the case of *linear functions*. There the diagnostic paradigm uses linear models for $g_k(\mathbf{x}) = \log p(k \mid \mathbf{x})$ which in the case of two classes reduces to logistic regression. For classes 0 and 1 the deviance (the difference between twice log likelihood and its maximum) is

$$D(\theta;\mathcal{T}) = 2\sum_{p=1}^{n}\left[y^p \log \frac{y^p}{p(1 \mid \mathbf{x}^p)} + (1-y^p)\log\frac{\{1-y^p\}}{\{1-p(1 \mid \mathbf{x}^p)\}}\right]$$

This form has frequently been suggested in the neural network literature (Solla *et al.*, 1988; Hinton, 1989; Bridle, 1990; Spackman, 1992; van Ooyen & Nienhuis, 1992) from both likelihood and entropy principles. Surprisingly, it is common there to use a logistic model for *each* class, which does not ensure that the probabilities sum to one. The log-linear approach gives rise to what Bridle (1989, 1990) terms *softmax* and expressed equivalently as

$$p(k \mid \mathbf{x}) = \frac{\exp g_k(\mathbf{x};\hat\theta)}{\sum_j \exp g_j(\mathbf{x};\hat\theta)} \tag{2}$$

and in any case the classifier chooses the class maximizing $g_k(\mathbf{x};\hat\theta)$.

The sampling-paradigm model of normal distributions for the classes with different means but a common variance matrix gives rise to a log-linear model. However, the parameters are estimated from the joint distribution $p(\mathbf{x}, y)$ whereas logistic regression uses the conditional distribution $p(y \mid \mathbf{x})$. Efron (1975) showed that the information loss in ignoring the marginal distribution of $\mathbf{X}$ is appreciable asymptotically. (Later work is discussed by McLachlan, 1992, §8.5.) Remarkably, for this model the plug-in estimate from the sampling paradigm can be found by linear regression. For two classes it has been known since Fisher (1936) that regression of the indicator of class 1 on $\mathbf{X}$ gives a linear rule which can be thresholded to give the plug-in sampling-paradigm classifier. (The threshold is not in general $1/2$ but very close to it: Ripley, 1993b; Ripley & Hjort, 1994.)

For more than two classes Breiman & Ihaka (1984) developed this approach, and their work has been extended by Hastie, Tibshirani (see their papers in this volume) and by myself. This shows that the predicted values of linear regression of the class indicators for the K classes on $\mathbf{x}$ can be used to reduce the space $\mathcal{X}$ to the space $\mathcal{X}^*$ of at most $K-1$ dimensions, and within $\mathcal{X}^*$ classification can be done either via a Mahalanobis distance to the class means or via another distance to the targets $\mathbf{t}_k$. (In each case there are constants to be subtracted for each class before minimizing.) Thus an inappropriate linear model for $f(\mathbf{x})$ is compensated for by using a different classification procedure to give exactly the sampling-paradigm estimate of the log-linear model. (Unfortunately this is often not done in the pattern recognition literature where Euclidean distance from the fitted values to the targets is used.)

This suggests two useful procedures to employ flexible families $f(\mathbf{x};\theta)$:

1. Model the log-probabilities and classify by the class with the largest coordinate. Fit the model using softmax and maximizing the conditional likelihood

$$\ell(\theta;\mathcal{T}) = \prod_p p(y^p \mid \mathbf{x}^p)$$

possibly ‘penalized’ by a regularizing term such as the log prior probability.

2. Model $E(I(Y=k) \mid \mathbf{X} = \mathbf{x})$, fit by (possibly penalized) least-squares but classify by a distance in the space of predicted values. We can think of this as choosing a transformation of the space $\mathcal{X}$ to a space in which we use linear methods. (It allows the direct use of regression methods and software.)

Note that the distance in the second procedure is *not* the Euclidean distance to the targets and does not correspond to choosing the largest component, which is the common practice in the neural networks literature. The linear case suggests that the second of these procedures may be preferable (but see below).

The predictive approach

In the predictive approach we compute

$$p(k \mid \mathbf{x};\theta) = \frac{\exp g_k(\mathbf{x};\theta)}{\sum_j \exp g_j(\mathbf{x};\theta)}$$

and then average over the posterior distribution of θ given the training set $\mathcal{T}$. For a neural network this is unknown, but if $E(\theta;\mathcal{T})$ denotes the sum of the log-likelihood and the log-prior (the regularizing term), and H is its Hessian at a local minimum, then locally the posterior density is proportional to

$$\exp -E(\theta) \approx \exp -\left[E(\hat\theta) + (\theta-\hat\theta)^T H(\theta-\hat\theta)/2\right]$$

and integration over this peak suggests weighting the local minimum proportionally to

$$|H|^{-1/2}\exp -E(\hat\theta) \tag{3}$$

So one approximation is to find as many local minima as possible, and weight the $p(k \mid \mathbf{x}; \hat{\theta})$ proportional to (3). We may also want to consider the variation of $p(k \mid \mathbf{x}; \theta)$ across a peak. A simple idea is to note that approximately we have

$$\theta \sim N(\hat{\theta}, H^{-1}),$$

so we can sample from this distribution and average the predicted $p(k \mid \mathbf{x}; \theta)$. It makes sense to sample proportionally to the weight given to the peak, the approach taken in our examples.

2.1 Fitting neural networks *via* softmax

Since using softmax with neural networks is not in most texts, we give the details here to show that it is only slightly more complicated than the usual approach of fitting separate logistic regressions for each class versus the rest. We will consider a feed-forward neural network with *linear* output units providing a output y_k for each class.

The measure of fit is

$$E = \sum_k t_k \log \frac{t_k}{p(k \mid \mathbf{x})}$$

summed as usual over patterns. The targets t_k will usually be one for the correct class, and zero for the others, and the probabilities are given by (2) computed from (y_k).

To use back-propagation, or any other optimization method that uses derivatives, we need to find $\delta_k = \partial E / \partial y_k$, where y_k denotes $g_k(\mathbf{x}; \theta)$. It follows immediately that

$$\delta_k = \left(\sum_j t_j \right) p_k - t_k, \qquad p_k = \frac{\exp y_k}{\sum_j \exp y_j} \tag{4}$$

This is the initial step needed for back-propagation. Note that the outputs (y_k) can be shifted by a constant without altering the fitted probabilities. The most efficient way to avoid this over-parameterization is to drop one class (say class one) and set its output to zero, but include it in the sums in (4). This will, however, cause difficulties with the use of a weight-decay penalty.

3 Robustness

A strong warning is needed is that the theory only applies if the models and data are correct. Even one incorrect classification in the training set can cause havoc in fitting $\mathbf{f}(;\theta)$. I strongly advocate the use of robust methods of fitting (e.g. Staudte & Sheather, 1990). Some ideas for models for mis-classification are given by Copas (1988) and its discussion. In particular, procedure 2 above is only reasonable if the distributions in the space $\mathcal{X}^*$ are approximately normal with the same covariance matrix, and a robust estimate of the covariance matrix (e.g. Rousseeuw & van Zomeren, 1990) is needed to avoid a single bad fit grossly distorting the process.

Another area where robustness is needed is in choosing the complexity of the model; it is easy to choose complex models to fit cases which are actually wrongly classified. (This applies also to the predictive approach, which will give too much posterior weight to complex models.)

4 Tree-structured classifiers

Another approach which can also approximate $\mathbf{f}(\mathbf{x})$ and the Bayes classifier arbitrarily well in large samples is to design a partition $\mathcal{P}$ of $\mathcal{X}$, and to approximate the posterior probabilities (and hence the Bayes rule) within each cell of the partition by the frequencies of the classes of the training case which fall within that cell. Because general partitions are hard to work with, most procedures in this category use *decision trees* to specify the partition. These successively divide the space $\mathcal{X}$ and the training set on the levels of an attribute. The best-known approach within statistics is that of Breiman *et al.* (1984); there are also strands within the machine learning community (Dietterich, 1990; Quinlan, 1986, 1990, 1993) and in engineering (Chou *et al.* 1989; Chou, 1991; Gelfand & Delp, 1991; Safavian & Landgrebe, 1991).

These approaches differ principally in how they choose the model complexity, the size of the tree. Breiman *et al.* (1984) and Chou *et al.* (1989) use cost-complexity pruning, which is a backwards selection process with a penalty analogous to Mallow's C_p in regression (which penalizes the residual sum of squares by a constant times the number of parameters). Others (Crawford, 1989; Gelfand & Delp, 1991) prune with respect to an 'honest' measure of error rate. Buntine (1992) allows restricted n-ply lookahead and follows an approximation to the full Bayesian approach by averaging posterior distributions over many plausible trees.

One difficulty with tree methods is to allow a sufficiently large class of attributes on which to split; approximating partitions by splits along the axes may need very many cells. Allowing linear combinations of variables goes back at least to Henrichon & Fu (1969), and is part of the CART methodology of Breiman *et al.* (1984). Later authors have taken this up, sometimes under the name of *perceptron trees,* and non-linear functions are now proposed, for example small neural networks by Strömberg *et al.* (1991) and Guo & Gelfand (1992). These ideas replace a combinatorial choice over features and split-points by a continuous optimization over parameters of a (linear or non-linear) function.

Tree-classifiers have also been related to neural nets by Brent (1991) and Sethi (1990, 1991). They had essentially the same idea. Suppose we have 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. Sethi proposes replacing the threshold

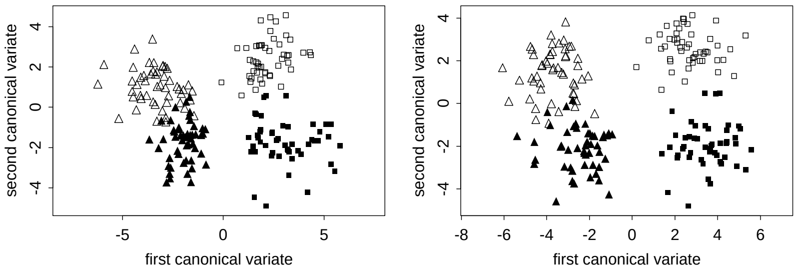

Fig. 1. Canonical variate plots for the *Leptograpsus* crab data. **Left** on original variables and **right** on $\log_{10}$ scale. The blue species is shown by triangles, the orange species by squares, and the symbols for females are filled.

nodes by logistic ones and re-training the network.

5 Examples

We include 3 small examples chosen to reflect some of the difficulties of relating our theoretical considerations to practice. A fixed weight-decay parameter chosen from past experience was used for each neural-network method.

5.1 *Leptograpsus* Crabs

Campbell & Mahon (1974) studied rock crabs of the genus *Leptograpsus*. One species had been split into two new species, previously grouped by colour form, orange and blue. Data are available on 50 specimens of each sex of each species. Each specimen has measurements on the width of the frontal lip, the rear width, length along the midline and the maximum width of the carapace and the body depth.

The data are plotted on the first two canonical variates in figure 1. (The canonical variates are the variables produced by a linear discrimnant analysis, and successively best separate the group means.) As the measurements are scales, we also considered taking logarithms, and as the figure shows this does increase slightly the separation between the groups. As there are four groups, the canonical variates span three dimensions, and on log scale the ratios of between- to within-group standard deviations are 25.5, 16.9 and 2.9 on the three variates. Thus the first two show 99.1% of the variation between the groups. Clearly the first canonical variate expresses the difference between the species, the second that between the sexes. This is almost a textbook example of multivariate normal populations with equal covariances. Applying that model and classifying 5,000 simulated examples estimated the Bayes risk at 1.5%, with errors between species having rate less than 0.1%. The k-nearest neighbour method estimated a lower bound for the Bayes risk as 1.4%.

The 200 examples were divided as 20 of each class for a training set and the remaining 120 as a test set. To make this a less standard problem, we also considered ignoring the colour forms and classifying on sex. Table 1 gives the number of errors (out of 120). (The statistical significance of these results is discussed in section 6.) In this example a classification tree is hopeless unless allowed to take linear combinations of measurements. All the neural networks considered had skip-layer connections and had two units in the hidden layer. Maximum-likelihood fitting was used with softmax or logistic output units (there being no difference for two classes); we also tried a regression neural network and classification by the nearest target and after post-processing by linear discriminant analysis.

Table 1. Error rates (out of 120) for the *Leptograpsus* crabs example. The three columns of results refer to first predicting both sex and colour, predicting sex from fully classified samples and then predicting just sex with colour unknown.

methods	4-way	sex only	sex only
colour given	yes	yes	no
linear discriminant	8	8	8
linear discriminant on log variables	4	4	4
quadratic discriminant	11	9	8
quadratic discriminant on log variables	9	7	7
neural network	6	5	2
linear, nearest target	12	12	8
post-processed by LDA	19	15	–
softmax on log variables	3	3	3
predictive	6	5	3
PP regression (4 ridge functions)	10	3	6
BRUTO	8	8	7
MARS (degree = 1)	10	8	4
MARS (degree = 2)	12	10	7
MARS (degree = 5)	19	15	7

BRUTO (Hastie & Tibshirani, 1990) is a specific method of fitting a smooth additive model, which is able to choose any smooth transformation for each variable, hence a logarithm. However, it does not know to choose the same transformation for each variable, and here differs negligibly from linear discrimination. In all the examples the PP regression, BRUTO and MARS methods used regression programs and option 2. above.

Note that despite the expectation from theory, we do better to classify the results of a neural network regression by Euclidean distance than via post-processing. To see why, consider a fit which occurred three of the five times when fitting the four-way classification, which had 12 errors using Euclidean distance and 23 errors via post-processing. The increase in errors comes from classifying one third of the blue males as orange females, a very rare confusion in other methods. Figure 2 shows

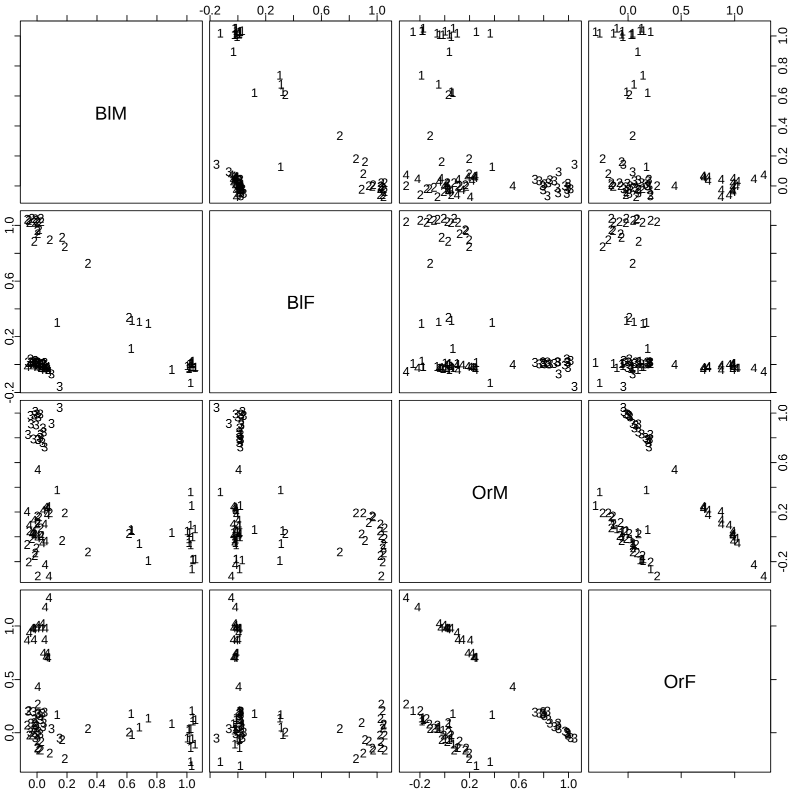

Fig. 2. Scatterplots of the outputs from a neural-network regression fitted to the training set of the crabs data. The groups labelled as 1 to 4 are in the order of the labels on the diagonal, blue males, blue females, orange males and orange females.

why; the within-groups covariance matrices are far from equal. The sample size is too small for the robust version of the method to be able to help much; it still has 20 errors.

5.2 Forensic glass

This set of data was collected by B. German on 214 fragments of glass, and appears in Murphy & Aha (1992). Each has a measured refractive index and chewmical composition (weight percent of oxides of Na, Mg, Al, Si, K, Ca, Ba and Fe). The fragments were originally classed as seven types, but some are infrequent or absent, and I re-grouped 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 analysis. The whole dataset was split randomly, resulting in a training set of size 89 and a test set of size 96. The analysis here overlaps that in Ripley (1994a) and uses the same training and test sets.

The results are given in table 2. The estimated lower bound for the Bayes risk was 11%, which no method comes close to achieving. (Of course, the bound may be too conservative.) The main neural network results used softmax and maximum-likelihood estimation. (They are averaged over 5 runs using different random initial weights.) Using neural network regression on the class indicators and post-processing by discriminant analysis took about twice as long, since the weight estimation for the network was slower. (The results were also considerably more variable than the log-linear fits.) 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. (Differences of 4% or less are not significant[1]; see section 6.)

Table 2. Error rates (%) for the forensic glass example for a test set of size 96. The two columns of results refer to all errors and to errors ignoring those between window non-float and window float glass.

methods	error rate %	
linear discriminant	41	22
nearest neighbour	26	17
neural network with 2 hidden	38	16
ditto, predictive	38	14
neural network with 6 hidden	33	16
ditto, predictive	28	12
ditto logistic outputs	39	17
ditto, via regression	41	19
PP regression (5 and 2 ridge functions)	40	19
BRUTO	42	22
MARS (degree = 1)	37	17
MARS (degree = 2)	31	19
classification tree	28	15

It is sometimes claimed that classification trees have an ease of interpretation. Figure 3 is none too easy to interpret. One difficulty in this example is that there is strong collinearity in the chemical measurements which sum to approximately 100%; it might be better to use a perceptron tree method (such as CART) but I do not have access to suitable software.

The predictive results are based on 10,000 random samples from Gaussians about

[1] Here and elsewhere the conventional significance level of 5% is assumed.

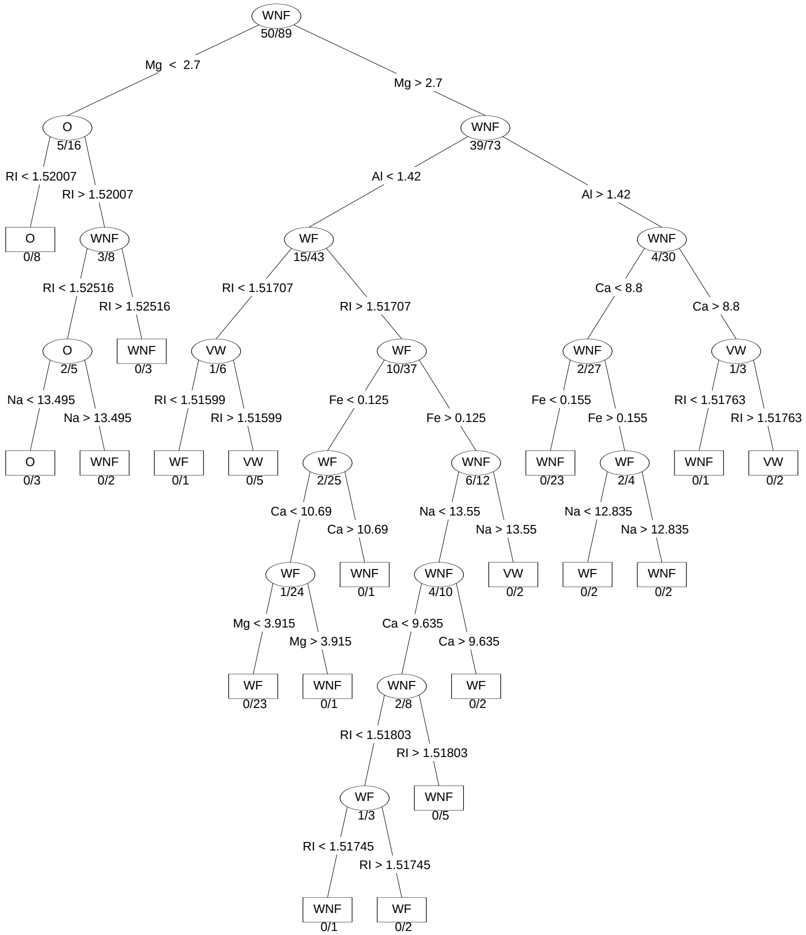

Fig. 3. Classification tree for the forensic glass example. At each node the majority classification is given, together with the error rate amongst training-set cases reaching that node.

the peaks found in 20 optimization runs; in fact many fewer would have sufficed. Although for two hidden units all 20 runs gave the same confusion matrix, the cases in error differed from run to run.

5.3 Low birth weights

Hosmer & Lemeshow (1989) give a dataset on 189 births at a US hospital, with the main interest being in low birth weight. The following variables are available:

`low`	birth weight less than 2.5kg
`age`	age of mother in years
`lwt`	weight of mother (lbs) at last menstrual period
`race`	white / black / other
`smoke`	smoking status during pregnancy (0/1)
`ptl`	number of previous premature labours
`ht`	history of hypertension (0/1)
`ui`	has uterine irritability (0/1)
`ftv`	number of physician visits in the first trimester
`bwt`	actual birthweight (grams)

Although the actual birth weights are available, we will be concentrating on predicting if the birth weight is low from the remaining variables. The dataset contains 5 pairs of rows which are identical apart from the ID; it is possible that these refer to twins but identical birth weights seem unlikely, so one of each has been deleted. Fifty cases were held back as a test set. The whole data set is discussed in Venables & Ripley (1994).

The conventional statistical approach would be to fit a logistic regression, allowing interactions between the categorical variables, and between the numerical variable age and lwt and categorical variables. Selecting and fitting a logistic regression on the training set produced error rates of 6/13 amongst the low-weight births and 5/37 amongst the normal births in the test set. A neural network with skip-layer connections and two and six hidden units both produced error rates of 4/13 and 8/37 in the two groups. Note that although the overall error rate is higher, more of the risk group are correctly identified.

Table 3. Error rates (%) for the low birth weight data. That of the normal group is given first, then the low weight group. Values in parentheses are standard errors.

methods	raw rate	smoothed
logistic regression	14 (6), 46 (14)	27 (4), 56 (8)
neural net (2 hidden)	22 (7), 31 (13)	28 (5), 34 (9)
multiple minima (2 hidden)	24 (7), 31 (13)	33 (4), 38 (8)
plus local averaging	24 (7), 38 (14)	34 (4), 36 (6)
neural net (6 hidden)	22 (7), 31 (13)	24 (5), 28 (11)

These numbers are too small to be reliable estimates of error rates. We can use the predicted posterior probability outputs from the models to produce better estimates (see section 6). As table 3 shows, these reveal that on this occasion we are lucky; there are several posterior error probabilities between 0.4 and 0.5 but none between 0.5 and 0.6.

The actual birthweights are available in this dataset. It might be thought desirable to train a regression to predict the actual birth weight and then threshold the result. In fact this proved to be worse. A regression model with interactions had error rates of 7/13 and 6/37, and a neural network had error rates of 8/13 and 4/37.

Predictive approach

We can also consider the predictive approach. For the logistic regression it will make little difference to the eventual classification, as the only non-linear element is the logistic itself, and that is linear around $x = 0.5$. Thus averaging over an (approximate) Gaussian distribution for the parameter estimates will bring in slightly the extreme estimates of posterior probabilities, but has a negligible effect on the estimated error rates.

For the neural networks this is no longer the case, not least because the optimization procedure finds many local minima. We experimented with 2 hidden units. Each of 20 runs produced a separate local minimum, and the weights given to each were 43%, 17%, 9%, 8%, 7%, 6%, 4%, $\ldots$. The effect of local averaging is also shown, based on a total of 1000 samples of weight sets.

The results here are too variable to draw useful conclusions about the value of attempting the integrations of the predictive approach.

6 Statistical Analysis of Comparative Experiments

Comparative experiments such as those in section 5 are very common throughout the neural networks literature and in related fields (e.g. Weiss & Kulikowski, 1991). It emerged at Les Arcs that the statistical techniques to design and analyse such experiments were much less well known to the participants.

The design and analysis of comparative experiments has been developed over most of this century, originally for agricultural experiments in the 1920's and 1930's and more recently for industrial and medical experiments. (Medical experiments are known as clinical trials.) An excellent basic reference is Box, Hunter & Hunter (1978). Experiments in our field are computer experiments and have much in common with work in the field of simulation; Kleijnen & van Groenendaal (1992) provide a non-technical introduction in that context which is amplified in Kleijnen (1987). Important ideas from that field include importance sampling and stratified sampling, both of which can be used to design 'difficult' test sets and to compensate for the increased difficulty in estimating error rates. For example, we might arrange to rare patterns to be well-represented in the test set, but down-weighted (e.g. Candela & Chellappa, 1993).

It should always be possible to give some idea of the variability of a quoted performance estimate. For example, if the error rate is (as usual) found by counting R errors in a test set of size N, then R has a binomial(N, P) distribution, and we can estimate the standard error of $\hat{p} = R/N$ by $\sqrt{\hat{p}(1-\hat{p})/N}$. This can be quoted in terms of error bars, for example as in Ng & Lippmann (1991) (shown

by Dr. Lippmann at the ASI). The convention in statistics is to quote approximate 95% confidence intervals, that is plus and minus *twice* the standard error. However, the crucial observation is that since the *same* test set was used for each method, the comparisons between methods are usually much more accurate than the error bars suggest. (In the terminology of the design of experiments we have a paired comparison, or a blocked experiment if there are more than two methods.) Consider the first two lines of table 1. As frequently happens, the 4 errors made in the second line are also made in the first line, so the difference in error rates is 4/120. Its standard error would be estimated as $\sqrt{(4 - 120/30^2)}/120 \approx 2/120$ so the difference is just significant. On the other hand the standard error of the unpaired difference between 4/120 and 8/120 is $\sqrt{0.0163^2 + 0.0227^2} \approx 3.37/120$ so the Ng-Lippmann approach would conclude no significant difference. A general version of this argument is to compare R_1/N to R_2/N and to assume that R_2 contains all R_1 errors. The paired standard error for $R_2 - R_1$ is then approximately $\sqrt{R_2 - R_1}$, so we can find a significant difference only with $|R_2 - R_1| \geq 4$, and this applies only if the worse method makes all the errors of the better method. This should be borne in mind in assessing tables 1 and 2.

The analysis of the last paragraph is approximate, in particular depending on approximating binomial distributions by normals. More precise methods are available, such as *McNemar's test* (Fleiss, 1981; Kotz *et al.*, 1985). Let n_A and n_B be the number of errors made by method A and not method B, and *vice versa*. Then McNemar's test (with continuity correction) refers

$$\frac{|n_A - n_B| - 1}{\sqrt{n_A + n_B}}$$

to a $N(0, 1)$ distribution, and an exact test refers n_A to a binomial $(n_A + n_B, 1/2)$ distribution. This suggests that we need $|R_2 - R_1| \geq 5$ for a significant difference.

One pitfall to be avoided is to give too much emphasis to statistically significant results. In an experiment in which method A with error rate 29.8% is significantly better than method B with error rate 30.1%, it is clear that the difference is extremely unlikely to be of practical importance, especially if we estimate the Bayes risk as 6%.

It is often possible to provide more accurate estimates of the true error rate than the observed error rate on the test set. The true error rate is given by

$$E\left[1 - \max_j p(j \mid \mathbf{X})\right]$$

If the classification method provides estimated posterior probabilities $\hat{p}(j \mid \mathbf{x})$ then

$$B = \frac{1}{\text{size of test set}} \sum_{p \in \text{test set}} \left[1 - \max_j \hat{p}(j \mid \mathbf{x}_p)\right]$$

provides an approximately unbiased estimator of the error rate if the posterior probabilities are accurate estimates. These estimates will be less variable because they average over the distribution of Y given $\mathbf{X} = \mathbf{x}$; see table 3. Devijver & Kittler

(1982, §10.7) call this the *risk-averaging approach.* and McLachlan (1992, §10.5) refers to *smoothing of the error rate*. The smoothed versions are particularly well-suited to paired comparisons and to use in bootstrapping. It is necessary to check that the model for the posterior probabilities is *well calibrated*, so that, say, about 70% of events given probability 0.7 occur. (Techniques for checking calibration are given by Dawid, 1982, 1986.)

If the performance has to be estimated from the training set, there are a number of ways to estimate and (approximately) remove the bias of the observed error rate (or B) evaluated at the training set. McLachlan (1992, Chapter 10) discusses cross-validation, the jackknife, the bootstrap and refinements such as the double bootstrap and the '0.632 estimator'. If these are used in comparative experiments, there is some danger that their biases will differ between methods and so bias the comparisons.

The major difficulty with statistical studies is selection bias, sometimes known as the 'bottom-drawer effect'. Statistically significant events do occur by chance, and if enough are tried one will occur. It is hard to estimate how much is tried and discarded, especially from published studies. (It is often the case that the example in the paper is the only successful one of those the author tried!) Another danger is that some methods are 'tuned' more assiduously than others. The competitions which are beginning to emerge avoid some of these effects, but maybe we need to test methods with average rather than expert users to reflect their 'real-world' performance levels.

Acknowledgements

All of the examples were computed using the statistical system S-Plus. The code for BRUTO and MARS was from the S library `fda` available by ftp from `statlib@lib.stat.cmu.edu`. I am grateful to Trevor Hastie and Rob Tibshirani for providing a pre-release copy of their code and for illuminating discussions. Discussions with David MacKay encouraged me to include the numerical results on the predictive approach.

The final preparation of this paper was done whilst visiting the *Isaac Newton Institute for the Mathematical Sciences* in Cambridge, England.

References

My earlier papers Ripley (1993a,b, 1994a) explore these and other issues in greater depth with examples. Ripley & Hjort (1994) will cover this ground in depth.

Berger, J. O. (1985) *Statistical Decision Theory and Bayesian Analysis.* New York: Springer.

Bishop, C. (1991) Improving the generalization properties of radial basis function neural networks. *Neural Computation* **3**, 579–588.

Bishop, C. (1993) Curvature-driven smoothing: a learning algorithm for feedforward networks. *IEEE Trans. Neural Networks*.

Box, G. E. P., Hunter, W. G. and Hunter, J. S. (1978) *Statistics for Experimenters: an Introduction to Design, Data Analysis and Model Building.* New York: Wiley.

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

Breiman, L. (1993) Hinging hyperplanes for regression, classification and function approximation. *Trans. IEEE IT* **3**, 999–1013.

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, U.C. Berkeley Statistics Department.

Brent, R. P. (1991) Fast training algorithms for multilayer neural nets. *IEEE Trans. Neural Networks* **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.

Bridle, J. S. (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, CA: Morgan Kaufmann.

Buntine, W. L. (1992) Learning classification trees. *Statistics and Computing* **2**, 63–73.

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

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.

Candela, G. T and Chellappa, R. (1993) Comparative performance of classification methods for fingerprints. US National Institute of Standards and Technology report NISTIR 5163.

Chou, P. A. (1991) Optimal partitioning for classification and regression trees. *IEEE Trans. PAMI* **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. Inf. Theory* **35**, 299–315.

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. Roy. Statistic. Soc. B* **50**, 225-265.

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

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

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

Dawid, A. P. (1982) The well-calibrated Bayesian (with discussion). *J. Amer. Statist. Assoc.* **77**, 605–613.

Dawid, A. P. (1986) Probability forecasting. In *Encyclopedia of Statistical Sciences* eds S. Kotz, N. L. Johnson & C. B. Read, pp. 210–218. New York: Wiley.

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

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

Dietterich, T. G. (1990) Machine learning. *Annual Review of Computer Science* **4**, 255–306.

Efron, B. (1975) The efficiency of logistic regression compared to normal discriminant analysis. *J. Amer. Statist. Assoc.* **70**, 892–898.

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

Fleiss, J. L. (1981) *Statistical Methods for Rates and Proportions.* Second Edition. New York: Wiley.

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

Friedman, J. H. and Stuetzle, W. (1981) Projection pursuit regression. *J. Amer. Statist. Assoc.* **78**, 817–823.

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition.* 2nd edition. London: Academic Press.

Gelfand, S. B. and Delp, E. J. (1991) On tree structured classifiers. In Sethi & Jain (1991) pp. 51–70.

Geman, S., Bienenstock, E. and Doursat, R. (1992) Neural networks and the bias/variance dilemma. *Neural Computation* **4**, 1–58.

Girosi, F. and Poggio, T. (1990) Networks and the best approximation property. *Biological Cybernetics* **63**, 169–176.

Gish, H. (1990) A probabilistic approach to the understanding and training of neural network classifiers. In *Proc. 1990 IEEE Conf. Acoustics, Speech and Signal Proc., Albuquerque* pp. 1361–1364. New York: IEEE Press.

Guo, H. and Gelfand, S. B. (1992) Classification trees with neural network feature extraction. *IEEE Trans. Neural Networks* **3**, 923–933.

Hampshire, J. B. and Pearlmutter, B. A. (1990) Equivalence proofs for multi-layer perceptron classifiers and the Bayesian discriminant function. In *Proceedings of the 1990 Connectionist Models Summer School* eds D. S. Touretzky, Elman, T. Sejnowski & G. Hinton, San Mateo, CA: Morgan Kaufmann.

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

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

Henrichon, E. G. Jr and Fu, K. S. (1969) A nonparametric partitioning procedure for pattern classification. *IEEE Trans. Comput.* **18**, 614–624.

Hinton, G. E. (1989) Connectionist learning procedures. *Artificial Intelligence* **40**, 185–234.

Hosmer, D. W. and Lemeshow, S. (1989) *Applied Logistic Regression.* New York: Wiley.

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 Networks.*

Hwang, J.-N., Li, D., Maechler, M., Martin, D. and Schimert, J. (1992) Projection pursuit learning networks for regression. *Engineering Applications Artificial Intelligence* **5**, 193-204.

Jones, L. K. (1990) Constructive approximations for neural networks by sigmoidal functions. *Proc. IEEE* **78**, 1586–1589.

Kleijnen, J. P. C. (1987) *Statistical Tools for Simulation Practioners.* New York: Marcel Dekker.

Kleijnen, J. P. C. and van Groenendaal, W. (1992) *Simulation: A Statistical Perspective.* Chichester: Wiley.

Kotz, S., Johnson, N. L. and Read, C.B. (1985) *Encyclopedia of Statistical Sciences.* Volume 5. New York: Wiley.

MacKay, D. J. C. (1992a) Bayesian interpolation. *Neural Computation* **4**, 415–447.

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

MacKay, D. J. C. (1992c) The evidence framework applied to classification networks. *Neural Computation* **4**, 720–736.

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

Miller, J. W., Goodman, R. and Smyth, P. (1993) On loss functions which minimize to conditional expected values and posterior probabilities. *IEEE Trans. Inf. Theory*.

Murphy, P. M., & Aha, D. W. (1992). *UCI Repository of Machine Learning Databases* [Machine-readable data repository]. Irvine, CA: University of California, Department of Information and Computer Science. Available by anonymous ftp from `ics.uci.edu` in directory `pub/machine-learning-databases`.

Neal, R. (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, CA: Morgan Kaufmann.

Ng, K and Lippmann, R. P. (1991) A comparative study of the practical characteristics of neural network and conventional pattern classifiers. In *Advances in Neural Information Processing Systems 3*, eds R. P. Lippmann, J. E. Moody and D. S. Touretzky, pp. 970–976, San Mateo, CA: Morgan Kaufmann.

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

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

Park, J. and Sandberg, I. W. (1993) Approximation and radial-basis-function networks. *Neural Computation* **5**, 305–316.

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

Quinlan, J. R. (1986) Induction of decision trees. *Machine Learning* **1**, 81–106.

Quinlan, J. R. (1990) Decision trees and decision making. *IEEE Trans. SMC* **20**, 339–346.

Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Richard, M. D. and Lippmann, R. P (1991) Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation* **3**, 461–483.

Ripley, B. D. (1993a) 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 & Hall.

Ripley, B. D. (1993b) Neural networks and flexible regression and discrimination. In *Statistics and Images* ed. K. V. Mardia. *Advances in Applied Statistics*, Abingdon: Carfax.

Ripley, B. D. (1994a) Neural networks and related methods for classification (with discussion). *J. Roy. Statist. Soc. B* **56**.

Ripley, B. D. (1994b) Comment on: 'Neural networks: a review from a statistical perspective'. *Statistical Science* **9**.

Ripley, B. D. and Hjort, N. L. (1984) *Pattern Recognition and Neural Networks – A Statistical Approach.* Cambridge: Cambridge University Press.

Rousseeuw, P. J. and van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points (with discussion). *J. Amer. Statist. Assoc.* **85**, 633–651.

Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E. and Suter, B. W. (1990) The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Trans. Neural Networks* **1**, 296–298.

Safavian, S. R and Landgrebe, D. (1991) A survey of decision tree classifier methodology. *IEEE Trans. SMC* **21**, 660-674.

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

Sethi, I. K. (1991) Decision tree performance enhancement using an artificial neural network implementation. In Sethi & Jain (1991) pp. 71–88.

Sethi, I. K. and Jain, A. K. eds (1991) *Artificial Neural Networks and Statistical Pattern Recognition. Old and New Connections.* Amsterdam: North Holland.

Smyth, P. (1993) Probability density estimation and local basis function neural networks. In *Computational Learning Theory and Natural Learning Systems 2* eds T. Petsche, M. Kearns, S. Hanson and R. Rivest. Cambridge. MA: The MIT Press.

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

Spackman, K. A. (1992) Maximum likelihood training of connectionist models: comparison with least-squares back propagation and logistic regression. *Proceedings of the 15th Annual Symposium on Computer Applications in Medical Care, Nov. 1991*, pp. 285–289. New York: IEEE Press.

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

Staudte, R. G. and Sheather, S. J. (1990). *Robust Estimation and Testing.* New York: Wiley.

Strömberg, J. E., Zrida, J. and Isaksson, A. (1991) Neural trees—using neural nets in a tree classifier structure. *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 137–140. IEEE Press.

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

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

Wahba, G. (1990) *Spline Models for Observational Data.* Philadelphia: SIAM.

Wasserman, P. D. (1993) *Advanced Methods in Neural Computing.* New York: Van Nostrand Reinhold.

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, CA: Morgan Kaufmann.