# Ordinal Models for Risk Assessment

Mark Mathieson

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

ABSTRACT

Many financial problems involve the classifying items into classes which have a natural ordering. Examples include bond rating, credit scoring and equity risk rating. This paper discusses the special features of financial classification problems where the classes are ordered. The use of techniques designed specifically for ordered classes results in simpler classifiers, making it easier to interpret the factors that are being used to discriminate between the classes. We propose flexible classifier for ordered classes, based on a neural network. This new classifier is applied to the volatility rating of UK equities, where it results in improved prediction accuracy. The non-linear model facilitates understanding of the effects of fundamental factors on volatility.

This paper appeared in the proceedings of the UK Institute for Quantitative Investment Research (INQUIRE) Autumn Seminar 1995, Hertfordshire, UK. A more widely available reference to (a more technical version of) the material in this paper is:

Mathieson, M. J. (1995) Ordinal models for neural networks. In *Neural Networks in Financial Engineering,* eds A.-P. N. Refenes, Y. Abu-Mostafa and J. Moody (World Scientific, Singapore).

This research is in association with GMO Woolley Ltd.

# 1 Motivation

Many financial problems involve classifying a pattern of information into one of a number of pre-determined classes. Examples include:

1. **Bond rating**
   Attempt to predict the rating of a new bond before the official rating is available, using patterns consisting of quantitative information about the company. Here the bond is classified into one of the classes ordered from AAA to B−.

2. **Corporate or consumer credit scoring**
   Assign companies or individuals to classes according to their credit-worthiness using patterns of information from credit surveys and company accounts.

3. **Equity risk rating**
   Classify stocks into risk groups according to their return volatility, dividend payments, or prospect of failure.

4. **Asset allocation**
   Allocate assets to one of a number of funds to achieve the desired risk/return profile.

In the first three examples the classes have a natural ordering. In the bond rating problem the classes are the ratings in order of increasing risk of default. The ordering leads us to assume that companies issuing bonds rated AAA are more similar to those issuing AA bonds than those issuing B− bonds. We have an *ordinal* classification task. The asset allocation example is a *nominal* problem: there is no single natural ordering of the funds.

This paper discusses the special features of financial classification problems where the classes are ordered. We demonstrate that the use of techniques designed specifically for ordered classes results in simpler classifiers, making it easier to interpret the factors that are being used to discriminate between the classes. We propose a classifier for ordered classes, based on a neural network, that is sufficiently flexible to make use of non-linear relationships in complex problems. The classifier is applied to the volatility rating of UK equities, where it is used for prediction and as a tool for understanding the effects of fundamental factors on volatility. We consider some aspects of choosing a classifier of appropriate complexity, assessing performance and checking the accuracy of predictions. Throughout this paper the more technical details are consigned to the appendix, and further information is given by Mathieson (1995).

## 2 Classification into Ordered Classes

Suppose that patterns in a classification problem belong to one of $K$ classes. We number the classes $1, \ldots, K$ corresponding to their natural order if one exists (as in the bond rating example), and arbitrarily otherwise (as for asset allocation). Statistical decision theory motivates us to quantify the loss incurred in assigning to class $k$ a pattern belonging to class $j$ as $L(j,k)$. The most appropriate *loss function* for a particular application will clearly be that obtained by eliciting the relative costs of incorrect decisions from an expert user. This is a difficult and time-consuming task: in the examples above the losses may be real financial ones that are hard to quantify. Users often find it hard to be consistent in their estimation of relative costs.

In the absence of reliable information on relative costs, a natural approach for unordered classes is to treat every mis-classification as equally damaging. This gives the 0–1 loss function: $L(j,k)$ takes the values 0 when $j = k$ and 1 when $j \neq k$. Measuring the performance of a classifier using 0–1 loss is equivalent to simply considering the mis-classification error rate. However, in a bond rating task the cost of mis-allocating an AAA rated bond to the class AA will almost always be less than the cost of mis-allocating it to the class B$-$, so considering only the error rate obscures important information when comparing the performance of classifiers. For ordered classes, losses that increase with the absolute difference between the class numbers are more natural choices in the absence of better information. Examples include $L(j,k) = \mid j - k \mid$ and $L(j,k) = (j-k)^2$.

We write a pattern that forms the input to the classifier as a (random) vector $X$. The objective is to classify patterns into one of the $K$ classes in such a way that we incur a low expected loss. (There are other sensible objectives, such as minimizing the expected maximum loss, but these are not considered here.) Let the *posterior probability* that an observed pattern $x$ belongs to class $k$ be $p(k \mid x)$. The optimal classifier or *Bayes rule* which minimizes the expected loss (see for example Ripley, 1994b) will assign $x$ to the class

$$c(x) = \arg \min_{k=1,\ldots,K} \sum_j L(j,k)\, p(j \mid x). \tag{1}$$

That is, we assign $x$ to the class $k$ that minimizes the sum in (1). Under 0–1 loss, this reduces to choosing the class with maximum posterior probability $p(k \mid x)$, which is intuitively reasonable. When $L(j,k) = \mid j - k \mid$ the Bayes rule can be written in a form similar to the rule under 0–1 loss (see the appendix) but it avoids classifying an observation into a class where there is a high probability of making a serious error. This means that patterns will sometimes be assigned to a class that does not have maximum posterior probability. The next section uses a simple example to illustrate the difference between the Bayes rules under different loss functions.

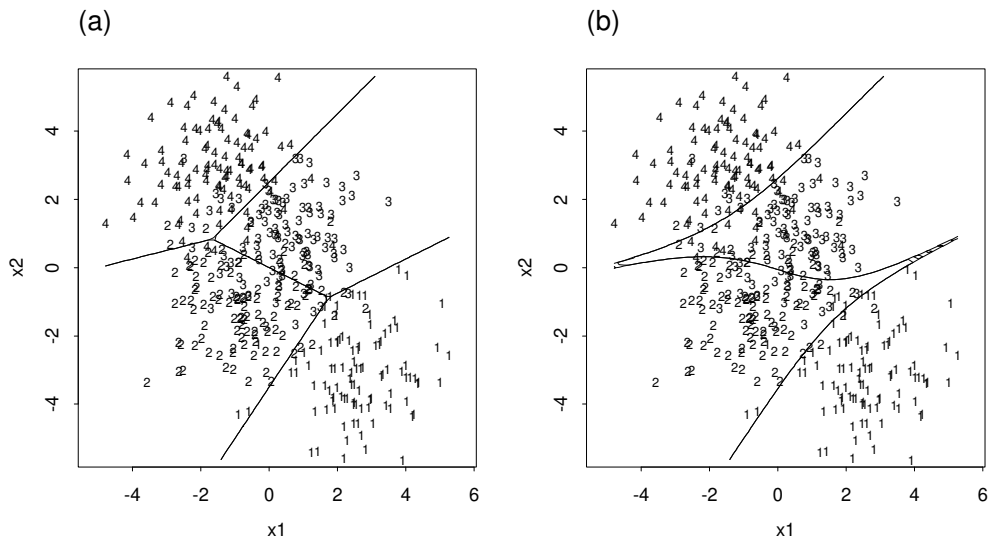


**Figure 1**: A synthetic example where patterns have just two components, $x_1$ and $x_2$. Patterns from class 1 are marked '1' etc. Decision boundaries are shown for the Bayes rule under **(a)** 0–1 loss **(b)** $L(j,k) = \mid j - k \mid$.

## 2.1 A Synthetic Example

We introduce a synthetic example that will be used to demonstrate a range of issues in ordinal classification. The example consists of patterns $x = (x_1, x_2)$ with two components, each pattern belonging to one of four ordered classes. To set the example in context, it may be useful to think of this as a rather optimistically simple model of equity risk where just two measurements of each stock, $x_1$ and $x_2$, are used as explanatory factors. For example, these might be historical measurements of borrowing ratio and earnings volatility. Figure 1 plots 100 patterns from each class. Each plotted point represents a different stock, and stocks with similar characteristics will lie close together. Every stock belongs to a risk class from one to four, where risk increases with increasing class number.

The example is synthetic and so the class probabilities $p(k \mid x)$ are known. We can therefore calculate Bayes rule exactly; the decision boundaries are shown in figure 1a for 0–1 loss, and in figure 1b for $L(j,k) = \mid j - k \mid$. Compare the positions of the decision boundaries in the region where class 2 meets class 4. Under 0–1 loss we aim to make a correct decision, and treat all errors as equal. However, when classifying $x$ under $L(j,k) = \mid j - k \mid$ a class may be given support by its position in the ordering, even if it does not have maximum posterior probability. Heuristically, it is better to mis-classify patterns in classes 2 and 4 as belonging to class 3 (incurring a maximum loss of 1) than to risk misclassifying class 2 as class 4 (or *vice versa*) and incur a loss of 2.

## 3 Predicting the Posterior Probabilities

If the posterior class probabilities $p(k \mid x)$ are known for each class $k = 1, \ldots, K$ then, by equation (1), this is sufficient to construct the optimal classifier. So any classification problem can be turned into one of predicting the posterior probabilities for any pattern that appears. The example of the last section was synthetic, so these probabilities were known. However, in real problems the posterior probabilities are unknown and so we construct a model for them. That is, we wish to present a pattern $x$ and have returned the probabilities $p(k \mid x)$ for each $k$. The model for $p(k \mid x)$ will be dependent on a set of parameters $\theta$, which are estimated to be $\hat{\theta}$ from a *training set* of $n$ correctly classified patterns. Substituting these estimates for $\theta$ in the model gives estimated posterior probabilities, which in turn we substitute in (1) to obtain the *plug-in* estimate of the Bayes rule.

Standard methods for nominal classification problems (reviewed by Fukunaga (1990) and Ripley (1995a) amongst others) can certainly be used to build the model for predicting the probabilities. However, if the classes in a classification problem are ordered we propose that this should be reflected in any model that is used. There are two main reasons for this:

1. The parameterization of the model will be more parsimonious than for comparable nominal models: the description of how to obtain the posterior probabilities from the patterns will be more concise. When there are two equally good explanations of a situation we prefer the simpler one, all other things being equal — an intuitive principle known as Occam's Razor. A more parsimonious model means that the variance of the estimated probabilities should be lower, and the approximations to the Bayes decision boundaries can be more accurate when limited training examples are available.

2. Interpretation of the model will be easier. It is easier to explain which features of the patterns are being used to discriminate between the classes using ordinal models than with comparable nominal models.

For ordered classes, we will model the cumulative class probabilities $\gamma_k = p(C \leqslant k \mid X = x)$, the probability that the class $C$ is less than $k$ when we observe a pattern $x$. The *ordinal logistic regression* (OLR) model for $K$ ordered classes models defines $\gamma_k$ by

$$\log\left(\frac{\gamma_k}{1-\gamma_k}\right) = \phi_k - \eta(x) \qquad (k = 1, \ldots, K-1), \tag{2}$$

where to ensure that the $\gamma_k$ are increasing we impose the constraints

$$\phi_1 \leqslant \ldots \leqslant \phi_{K-1} \tag{3}$$

on the *cut-points*. For the linear version (McCullagh, 1980) we take $\eta(x) = x\beta$, where $\beta$ is a column vector of the same length as the row vector $x$, giving parallel regressions

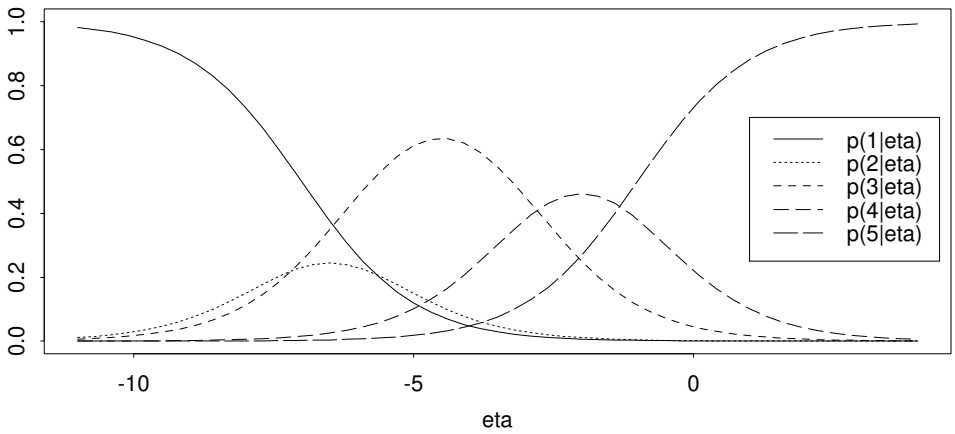


**Figure 2**: Class probabilities $p(k \mid \eta)$ plotted against $\eta$ under the ordinal logistic regression model. We have chosen $K = 5$ classes and $(\phi_1, \phi_2, \phi_3, \phi_4) = (-7, -6, -3, -1)$.

on the logistic scale. Rewriting (2) gives an expression for the posterior class probabilities

$$\begin{aligned} p(C = k \mid X = x) &= \gamma_k - \gamma_{k-1} \\ &= \sigma(\phi_k - \eta(x)) - \sigma(\phi_{k-1} - \eta(x)) \qquad (k = 1, \ldots, K) \end{aligned}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function, and for notational convenience we define $\phi_0 = -\infty, \phi_K = \infty$. Observe that the posterior probabilities depend on the patterns $x$ only through $\eta$, which increases with the predicted class of $x$. Figure 2 shows the posterior probabilities $p(k \mid \eta)$ plotted against $\eta$ for an example with five ordered classes.

The training set of patterns whose true classes are known is used to select values for the parameters $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_k)$ (called *cut-points*) and $\beta$. The parameters are estimated by maximum likelihood; that is we choose values to maximize the probability of observing the independent patterns in the training set. The likelihood depends on the sampling method used to obtain the training set, but in every common case the computational task is to maximize

$$\ell(\beta, \boldsymbol{\phi}) = \prod_{i=1}^{n} p(c_i \mid x_i) = \prod_{i=1}^{n} \left[\sigma(\phi_{c_i} - \eta(x_i)) - \sigma(\phi_{c_i - 1} - \eta(x_i))\right],$$

where there are $n$ training patterns and the pattern $x_i$ belongs to class $c_i$. Of course, we maximize while constraining the cut-points to be increasing as in (3).

Figure 3a shows the decision boundaries fitted under linear OLR; they are obviously inadequate approximations to the Bayes boundaries. In this simple case we can see by eye that there is sufficient information in the patterns to allow discrimination far better than we have achieved. (Ensuring that there is sufficient information to allow

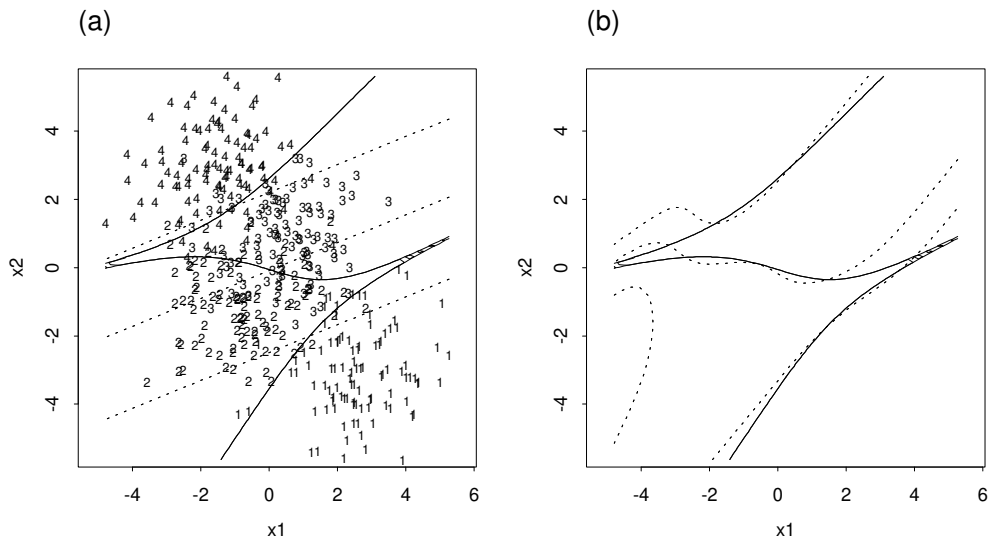


**Figure 3**: Training set for the synthetic example. Samples from class 1 are marked '1' etc. Decision boundaries for the Bayes rule under loss $L(j,k) = \mid j - k \mid$ (solid) and **(a)** OLR (dashed), **(b)** NOLR with 3 units (dashed).

discrimination that is adequate for our eventual purpose is a problem in feature selection that we do not consider.) In the next section we propose a new, more flexible, ordinal model.

### 3.1 A Flexible Model for Ordered Classes

OLR is rather inflexible; its decision boundaries will always be parallel lines and so it performs poorly even on our simple example. Financial problems are never this simple because the data that is used as patterns is far more noisy. In a bond rating task we cannot find information to use as patterns that separates the different ratings anywhere near as well as in our synthetic problem. This inflexibility is one of the reasons that ordinal models are not in widespread use in real problems. A method is required that accommodates departures from the linearity assumptions of OLR. We do this by letting $\eta(x)$ be the output of a neural network, rather than the linear function $\eta(x) = x\beta$. The result is non-linear ordinal logistic regression (NOLR).

For the purposes of this paper it is sufficient to think of the network as a flexible function from the input pattern vector $x$ to the output $\eta(x)$. Further details are given in the appendix, and by Mathieson (1995). A neural network is not the only non-linear function that could be used here, but there are technical advantages in doing so. The function formed by the network depends on the values of the network parameters (known as *weights*) in the same way that the exact function $\eta$ in linear OLR depends on $\beta$. We estimate the weights and cut-points by maximum likelihood just as we did for $\beta$. In neural network jargon this process of using known examples to determine the net-

work parameters is called *learning*, but this is exactly the same idea as estimating the parameters in a linear regression, for example. (It is more computationally difficult, partly because there are not explicit expressions for the parameter values that maximize the likelihood. The parameters therefore have to be estimated iteratively.)

One way of choosing the degree of flexibility of function $\eta$ that is formed by the network is by changing the number of *hidden units* in the network. The more hidden units the more complex the function; NOLR with no hidden units is equivalent to linear OLR. Figure 3b shows the decision boundaries fitted by NOLR with three hidden units; they are much better approximations to the Bayes decision boundaries than given by linear OLR.

# 4 Alternative Ordinal Models

In this section we discuss the advantages of the new NOLR model, and compare it with conventional methods for dealing with ordered classification problems. Many of these advantages are demonstrated in section 7 when NOLR is applied to the volatility rating of UK equities. Financial practitioners usually treat ordered problems in exactly the same way as nominal ones — even when an appropriate loss function is used (section 2), the class ordering is seldom incorporated into a probability model. This is partly due to the lack of good computer software for ordinal models, and partly because conventional techniques such as linear OLR are not flexible enough to model complex financial problems such as bond rating, equity risk rating and credit scoring.

The most common technique used with ordered classes is a regression of the integers $1, \ldots, K$ on the feature vectors $x$ in the training set. This may be a linear regression, or a non-linear regression using a flexible function such as a neural network. This technique was used by Moody and Utans (1995) to predict bond ratings; they employed a network to do a non-linear regression. It is not necessary to use the integers: any ordered *scores* $s_1 \leqslant \ldots \leqslant s_K$ can be used. The linear or non-linear function $\eta(x)$ is then fitted by least squares. That is, we minimize $\sum_{i=1}^{n} \left(s_{c_i} - \eta(x_i)\right)^2$ where training example $x_i$ belongs to class $c_i$. A slightly more sophisticated approach is to allow the scores to be variable, and estimate them also. The decision rule that is usually used is to assign a sample $x$ to the class corresponding to the score that is closest to $\eta(x)$. That is we assign $x$ to class $\arg\min_{k=1,\ldots,K} \mid s_k - \eta(x) \mid$.

These techniques, known as *scoring rules*, have serious practical disadvantages (in common with other methods that do not provide full probability models). The rather arbitrary decision rule does not carry the guarantee of optimality that comes with the Bayes rule. In addition, the probabilities predicted by NOLR can be used to refine the classification process. For example, when predicted probabilities are similar for all of classes, we can declare 'doubt' and re-examine the observation. Perhaps the pattern in question is a special case that needs consideration by an expert. This cannot be done with scoring rules.

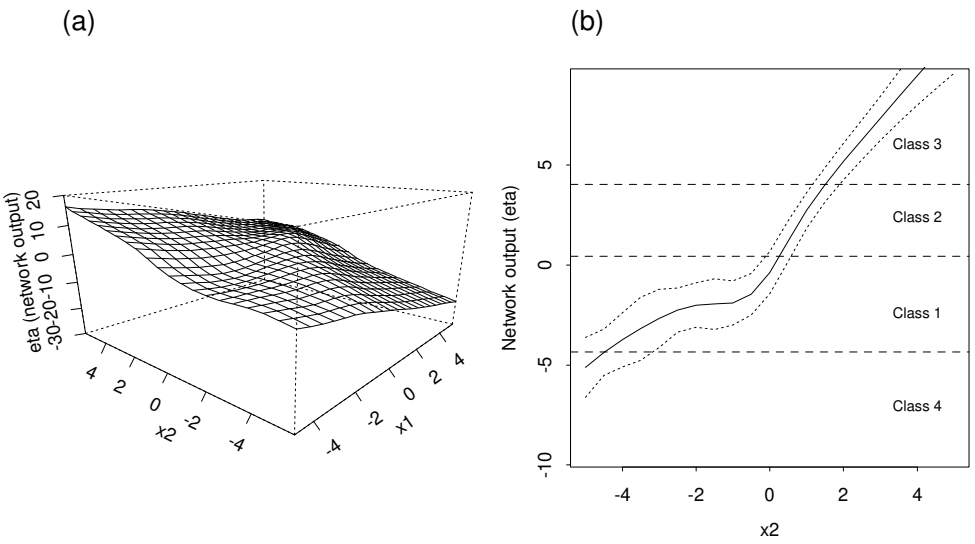


**Figure 4**: **(a)** Network output $\eta(x)$ plotted against the factors for NOLR with 3 hidden units. High values of $\eta(x)$ correspond to high predicted classes. **(b)** A section through (a) fixing $x_1 = -2$. Dotted lines are approximate 95% pointwise confidence intervals. Dashed lines are decision boundaries.

Financial practitioners generally decide that the advantage of using scoring rules is that they produce a variable that increases with the predicted class of $x$. This is useful for interpretation, because we can examine the way that this variable is sensitive to slight changes in the patterns. However, NOLR retains this advantage because the network output $\eta(x)$ also has this property. Furthermore, we can interpret $\eta(x)$ on a scale like the one shown in figure 2 which gives the probability of $x$ belonging to each class. Figure 4a illustrates how $\eta(x)$ changes as $x$ varies for our synthetic example. We can see that when factor $x_1$ is low and factor $x_2$ is high, the predicted class is high. From the discrete categories we have fitted a continuous measurement that increases with increasing class. The decision boundaries are just contours of this surface. Linear OLR fits a surface that is completely flat; notice that the slight curvature of the surface fitted by NOLR is sufficient to change the decision boundaries from those in figure 3a to those in figure 3b.

Suppose that the factor $x_1$ is fixed at $-2$. Then figure 4b shows how $\eta(x)$ changes as $x_2$ is varied. The general trend is to predict a higher class as $x_2$ increases. Interpretations such as this are extremely complicated with most nominal methods. The approximate 95% pointwise confidence intervals for $\eta(x)$ are obtained by local linearization (Seber & Wild, 1989), and can be converted to confidence intervals for the posterior class probabilities. Methods for measuring confidence in predictions with scoring rules are far less convincing.

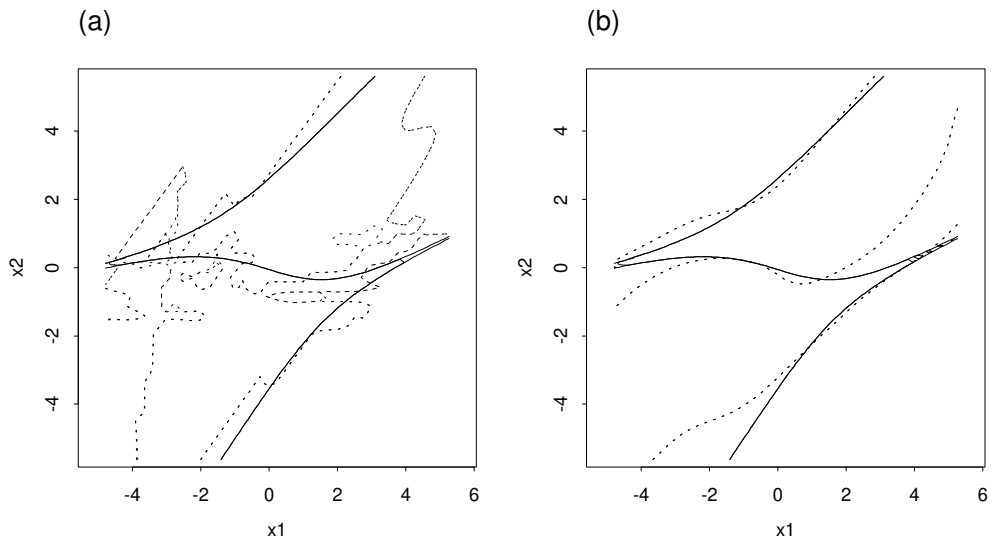


**Figure 5**: Decision boundaries for the Bayes rule (solid) and **(a)** NOLR with 12 hidden units (dashed) **(b)** NOLR with 12 hidden units and weight decay $\lambda = 0.05$ (dashed).

# 5 Choosing Model Complexity

In section 3.1 we demonstrated that the flexibility of the function defined by the network is increased by increasing the number of hidden units. NOLR with no hidden units gave the boundaries in figure 3a, and with three hidden units it gave the improved boundaries in figure 3b. If we continue to increase the number of hidden units to 12, for example, we obtain figure 5a. Now the decision boundaries are far too complex. The training set is being over-fitted: its random characteristics are being modelled rather than its general features. Recall that our objective in classification problems is to perform well on future (unseen) examples rather than to classify the training set with complete accuracy. In the context of bond rating, we are not usually interested in learning how to predict the rating of bonds whose rating we already know — we wish to *generalize* in order to predict the risk rating of new bonds.

We have moved from insufficient flexibility to classifiers that are potentially too flexible. In fact, the networks that we employ in NOLR form a family of functions that can be *arbitrarily* flexible. (They can approximate uniformly any continuous function on closed bounded regions.) There are many methods in statistics that can be used to choose models of appropriate complexity, but they are often ignored in reports of financial applications. Ad-hoc methods are adopted or, worse, over-fitted classifiers with little predictive power are used without checking. Extended discussion of methods for model choice is given by Ripley (1995b). Here we will restrict our attention to some of the more intuitive methods, and illustrate their application to NOLR.

**Cross-Validation**

One way of estimating future performance is to divide the training examples into $v$ parts, build a classifier using all but one of the parts, and then assess the performance using the remaining part. When this is repeated leaving out each part in turn, and averaging the performance results, it is known as *$v$-fold cross-validation* (Stone, 1974). We can take $v$ to be any number between 2 and $n$, the number of training examples. (Even if we can afford the time to build and test the large number of classifiers required for $n$-fold cross-validation, the balance between variability and bias of performance estimates suggests that this is not a good idea. Choosing $v \approx 5$ is usually a good compromise.) In fact, 4-fold cross-validation of the training set (using likelihood as a measure of goodness-of-fit) was the method we used to select 3 hidden units for NOLR, obtaining the classifier in figure 3b. (If instead we measure goodness-of-fit using loss-per-sample under $L(j,k) = \mid j - k \mid$, 4-fold cross-validation chooses a network with 4 hidden units.)

**Regularization**

Over-fitting of the training examples is often characterised by decision boundaries that have high curvature. For NOLR this is equivalent to choosing a network that gives a function $\eta(x)$ that is very curved. Networks tend to define functions that are not smooth when the weights are large and so we indirectly penalize functions which are not smooth by penalizing networks with large weights. This is termed *weight decay* and is discussed further in the appendix. The smoothing effect of weight decay on the synthetic example can be seen in figure 5. The over-fitted classifier in figure 5a is smoothed to that in figure 5b, which is a much better approximation to the Bayes rule. When regularization is used the number of hidden nodes is less critical.

Maximizing the likelihood as described in section 3 is equivalent to choosing weights $\mathrm{w}$ and cut-points $\boldsymbol{\phi}$ to minimize $-\log \ell(\mathrm{w}, \boldsymbol{\phi})$. Adding weight decay changes the minimand to $-\log \ell(\mathrm{w}, \boldsymbol{\phi}) + \lambda D(\mathrm{w})$ where $D(\mathrm{w}) = \sum_{i,j} w_{ij}^2$ and $\lambda > 0$. The trade-off is between the two terms: when the weights are large the sum of their squares $D(\mathrm{w})$ is large, but then the function is flexible and the fit is good, so $-\log \ell(\mathrm{w}, \boldsymbol{\phi})$ is small. Conversely, when the weights are smaller, the fit to the training set is worse. The larger the value of $\lambda$ the greater the smoothing effect; cross-validation was used to choose an appropriate value of $\lambda$ for the synthetic example in figure 5b. More details can be found in the appendix.

**An Information Criterion (AIC)**

AIC (Akaike, 1973) attempts to correct the bias in estimating the deviance (minus twice the log of the likelihood) of an independent test set of examples by the training set deviance. AIC chooses the probability model that minimizes $-2 \log \ell(\mathrm{w}, \boldsymbol{\phi}) + 2p$ where $p$ is the number of parameters in the model. (We assumes that fitting is by maximum likelihood and that the true model is contained within our flexible family of models.) In fact, AIC selected the NOLR model with 12 hidden units shown in figure 5a that is too complex.

| Classifier | Average loss on test samples |
|---|---|
| Bayes | 0.211 |
| OLR | 0.286 |
| NOLR (3 units) | 0.218 |
| NOLR (12 units) | 0.240 |
| NOLR (12 units and weight decay $\lambda = 0.05$) | 0.219 |

**Table 1**: Classifier performance for the synthetic example test data. An approximate standard error the average losses is $s = 0.015$, but see the text for more accurate comparisons.

## 6 Comparing Classifier Performance

When a number of different classifiers have been chosen it is necessary to compare their performance. An independent test set of 800 unseen patterns was used to assess the performance of various classifiers on our synthetic example. The results are given in table 1, where we list the average loss incurred by each classifier when it is used to assign the patterns in the test set to classes. Throughout we have used the loss function $L(j,k) = \mid j - k \mid$. Our target performance is that of the Bayes rule, which we can calculate since the example is synthetic.

Our test set is a random sample, and so we need to decide whether differences in the figures shown are attributable to random fluctuations in the test examples that we chose or due to genuine differences in performance. Using paired $t$-tests, the performances of both NOLR with 3 units and NOLR with 12 units plus weight decay $\lambda = 0.05$ are not significantly different from the Bayes rule. All other differences in performance are significant. More details are given in the appendix.

A concealed problem here is that computational techniques for finding values of the parameters that maximize the likelihood usually only find *local* not *global* maxima. Different local maxima will give rise to different decision boundaries, and we can never be sure whether we have found the global maximum. Even in our simple synthetic problem, many local maxima were found in the likelihood surface. As a crude approximation to a predictive Bayesian scheme (Aitchison & Dunsmore, 1975), the predicted probabilities were averaged over the models fitted by four runs of the optimizer from randomized starting weights.

## 7 Return Volatility Rating of UK Equities

In this section NOLR is applied to an ordinal classification problem in stock selection: the volatility rating of UK equities on the basis of fundamental information about the companies. The important practical features of NOLR are demonstrated, such as interpretability and flexibility. We concentrate on one time period for simplicity, but the

advantages in using the NOLR model are valid over extended periods or when used on multiple occasions.

Predicting the risk associated with individual stocks is a first step towards predicting — and so controlling — the risk involved in holding particular equity portfolios. We may believe that an investor's tolerance of increased risk is rewarded by high expected return (as suggested by equity pricing models), so a risk model could form part of an equity valuation scheme. In this case we can expect risk rating to be difficult in markets that have high efficiency. Perhaps the most important objective is to acquire some understanding of how fundamental factors affect risk, as this is an area where interactions are complex and intuition can be poor. A complete model of the relative risk of stocks would incorporate predictions of return volatility, company failure and changes in the dividend yield, for example. In this application we consider the return volatility component.

The dataset consisted of 675 stocks quoted on the London Stock Exchange on 30 June 1990 and during the following 12 months, with a market capitalization of at least £20m. Ten input variables were used, including proprietary factors thought to affect risk. Some of the variables were categorical, such as that indicating the presence or absence of intangible assets on the balance sheet. Financial and property companies were excluded from this analysis as their characteristics are distinct from the rest of the market and were thought to require separate modelling. The dataset contains only contemporary public information and minimizes survivor bias.

**Treatment as a Classification Problem**

Return volatility was quantified using a robust measure of the variance of the daily return (relative to the FT All-Share index) over one year of trading days. (Other measures, such as beta values and semi-variance, can be used in the same way.) The stocks were classified into quintiles 1–5 in order of increasing volatility, and the problem was treated as one of classification into these five ordered groups rather than of predicting volatility exactly. This is appropriate because identifying *extremes* of volatility is of particular interest. Further, when direct predictions of volatility are not required, treatment as a classification problem is advantageous since it only requires modelling of the decision boundaries, and so is likely to be less difficult than predicting actual volatility values. This framework also allows the easy extension of the model to predict an expert analyst's assessment of stock volatility on an ordinal scale.

**Performance Assessment**

We predict the volatility class of 175 randomly selected stocks using the remaining 500 as a training set, using loss $L(j,k) = \mid j-k \mid$ throughout. Without transformation of the input variables it is relatively easy to find network methods that incur lower losses on the test set than linear methods. This is partly the result of large differences in scale of the variables; inappropriate inputs to linear classifiers often underlie some

of the reported 'success stories' with neural networks. However, when appropriate transformations have been made there is still gain in applying non-linear methods to this problem.

Under NOLR, 10 hidden units were sufficient to classify in excess of 80% of the training set correctly, so this architecture was used for the selection of regularization parameters using 4-fold cross-validation on the training set. The NOLR model selected using loss-per-sample as a measure of fit obtains an average test set loss of 1.04 per sample, significantly better than OLR (1.18 per sample). Using likelihood as a measure of fit, a more complex model is fitted, with test set loss of 1.24 per sample. AIC (section 5) selects a model with more that 10 hidden units which is clearly too complex. For comparison, if we allocate samples to classes at random we expect loss of 1.60 per sample. The results of some other classification techniques, along with some more detailed comments, are given in the appendix. Although this is not intended to be an exhaustive comparison of techniques, NOLR certainly performs better than linear methods. *Softmax* (Bridle, 1990) is the nominal network method most similar to NOLR; it is non-linear, *nominal* logistic regression. Cross-validation using loss-per-sample as a measure of fit selects a model that experiences a test set loss not significantly different from NOLR. However, the important advantage of NOLR comes in the improved interpretability of the model that has been chosen; we discuss this below.

The input variables intentionally contained no direct measure of the historical volatility of a stock. Using the historical volatility class of a stock (based on the 12 months prior to 30 June 1990) as a prediction of the volatility in the following 12 months resulted in a significantly greater loss than using either of the non-linear methods on the (non-historic) input variables. This indicates that we have succeeded in identifying fundamental factors affecting volatility. An improvement in performance would be expected if a measure of stocks' past volatility is included in the model. This requires consideration of the time scale on which volatility should be measured, and the stability of the effect of fundamental factors on volatility.

**Interpretation of the Ordinal Model**

NOLR has important advantages over other nominal flexible methods since it provides a model that is more interpretable. Figure 6 shows how the volatility levels predicted by NOLR vary with the input variables *Quality* (a proprietary measure of investors' 'comfort' in holding a stock) and *Borrowing Ratio* (debt divided by total capital employed). In both cases, volatility increases as borrowing ratio increases. The interaction with quality is more complex; in figure 6, for low values of Borrowing Ratio, volatility decreases as quality increases. It is difficult to obtain an interpretation of this form using the five separate outputs of other network methods, such as softmax. Our volatility measure $\eta(x)$ is directly interpretable against a probability scale (such as in figure 2), and we can give confidence intervals for $\eta(x)$ (as in figure 4b) and for the estimated class probabilities $p(k \mid x)$.

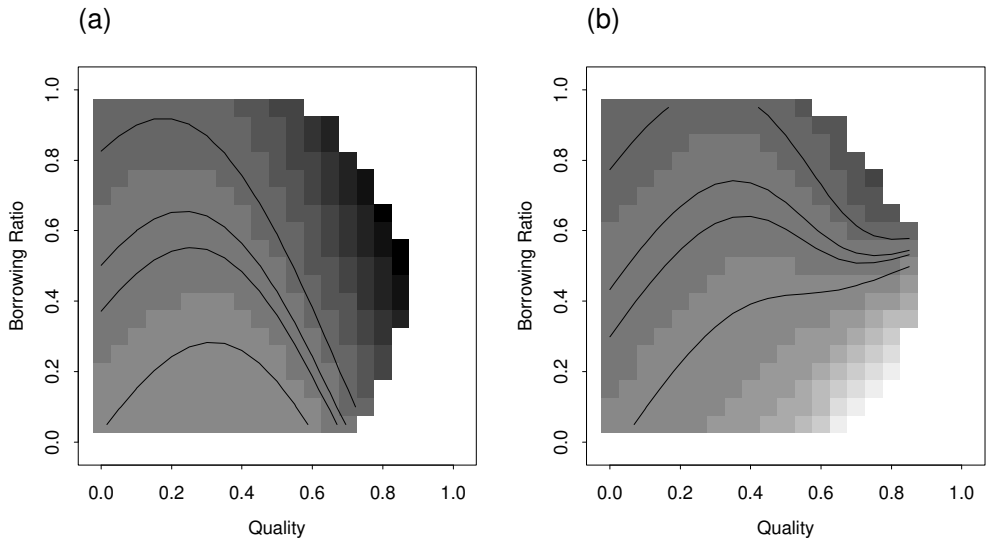


**Figure 6**: Profiles of predicted volatility as the two (rescaled) variables Quality and Borrowing Ratio vary. Darker shading indicates higher predicted volatility. The lines are the Bayes decision boundaries under loss $L(j,k) = \mid j - k \mid$. Plots (a) and (b) are formed by fixing the other variables at values in the centres of two clusters of companies (found by k-means clustering). Using cluster centres rather than setting variables to their average values is a simple way of trying to avoid 'empty' regions in the pattern space — the 'average' company is unlikely to exist. Predicted volatility is plotted on the convex hull of members of the training set.

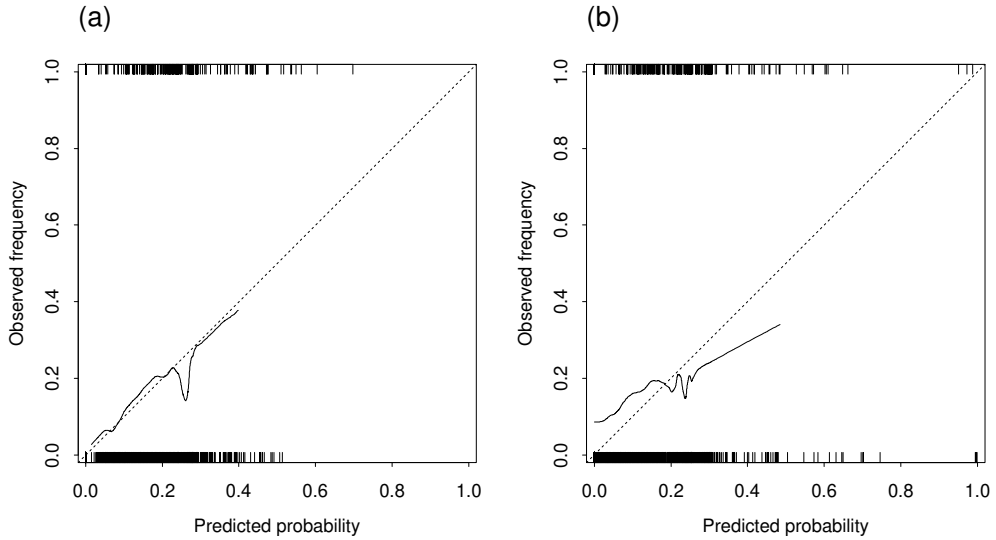


**Figure 7**: Calibration plots for the NOLR model on test set (observed frequencies against predicted probabilities). **(a)** The well-calibrated model. **(b)** The over-fitted model selected by AIC. This model is not well-calibrated. The predictions of high and low probabilities are over-confident, which is characteristic of over-fitting. Ticks on the axes denote predicted probabilities for $p(k \mid x)$ for $x$ in the test set; events on the upper axis occur, those on the lower axis do not.

## 8 Predicting Probabilities Accurately

As a check on the validity of the NOLR models that were fitted in the volatility rating problem, we examine the accuracy with which they predicted the probabilities of unseen stocks belonging to the assigned volatility group. A *well-calibrated* model is one where events that are predicted with a probability $p$ are observed to actually occur with long-term frequency $p$ (Dawid, 1986). Figure 7a shows the well-calibrated NOLR model fitted by cross-validation of deviance as described in the previous section. However, figure 7b shows the model that was too complex (fitted by AIC). The predicted probabilities deviate systematically from the long run frequencies. It is a characteristic of over-fitted models that predictions are over-confident for rare and frequent events. This could be dangerous in practice if, for example, we underestimate the probability that a stock has high volatility.

It is not *necessary* for a classifier to predict the posterior class probabilities accurately in order to perform well. A range of methods can be used to model the decision boundaries, and they do not have to predict posterior probabilities accurately or, indeed, at all. An example of such a classifier was the scoring model of section 4. However, it is certainly *sufficient* to be predicting probabilities accurately in order to classify with success — this is shown by equation (1).

If the model is to be interpretable then predicting probabilities accurately will certainly be important. Further, section 2 shows that we need to be predicting the class probabilities $p(k \mid x)$ accurately to be able to approximate the Bayes rule for *any* loss function with a given probability model. This may be useful if we are unsure of the appropriate loss function for an application, or suspect that it may be changed by the user once the model has been selected. We also need to avoid 'tuning' a classifier to a particular loss function. (This would occur if we minimized $\sum_{i,k} L(c_i, k)p(k \mid x_i)$ on the training set, for example.) Perhaps the loss function has a direct economic connection and will change over time. This is particularly relevant for ordinal classification problems.

## Acknowledgements

The author gratefully acknowledges productive discussions of this work with Professor B. D. Ripley. This research was funded by the UK Engineering and Physical Sciences Research Council and GMO Woolley Ltd, who provided the equity database and related information.

## References


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

Akaike, H. (1973) Information theory and an extension of the maximum likelihood principle. In *International Symposium on Information Theory,* eds B. Petrov and F. Cáski (Akademiai Kaidó, Budapest).

Bishop, C. M. (1992) Exact calculation of the Hessian matrix for the multilayer perceptron. *Neural Computation,* **4** 494-501.

Bridle, J. S. (1990) Probabalistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neuro-computing: Algorithms, Architectures and Applications,* eds F. Fogelman-Soulié and J. Hérault (Springer Verlag, New York).

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

Dawid, A. P. (1986) Probability forecasting. In *Encyclopedia of Statistical Sciences,* eds S. Kotz, N. L. Johnson and C. B. Read (John Wiley, New York).

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

Gill, P. E., Murray, W. and Wright, M. H. (1981) *Practical Optimization* (Academic Press, London).

Hastie, T. and Tibshirani, R. (1987) Non-parametric logistic and proportional odds regression. *Applied Statistics,* **36** 260–276.

Mathieson, M. J. (1995) Ordinal models for neural networks. In *Neural Networks in Financial Engineering,* eds A.-P. N. Refenes, Y. Abu-Mostafa and J. Moody (World Scientific, Singapore).

McCullagh, P. (1980) Regression models for ordinal data. *Journal of the Royal Statistical Society Series B,* **42** 109–142.

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

Moody, J. and Utans, J. (1995) Architecture selection strategies for neural networks: application to corporate bond rating prediction. In *Neural Networks in the Capital Markets,* ed. A.-P. N. Refenes (John Wiley, New York).

Ripley, B. D. (1994a) Neural networks and flexible regression and discrimination. In *Statistics and Images 2,* ed. K. V. Mardia, *Advances in Applied Statistics* (Carfax, Abingdon).

Ripley, B. D. (1994b) Flexible non-linear approaches to classification. In *From Statistics to Neural Networks. Theory and Pattern Recognition Applications,* eds V. Cherkassky, J. H. Friedman and H. Wechsler (Springer Verlag, New York).

Ripley, B. D. (1995a) *Pattern Recognition and Neural Networks* (Cambridge University Press, Cambridge).

Ripley, B. D. (1995b) Statistical ideas for Selecting Network Architectures. In *Proceedings of the 3rd SNN Workshop, Nijmegen,* (Springer Verlag, London).

Seber, G. A. F. and Wild, C. J. (1989) *Nonlinear Regression* (John Wiley, New York).

Stone, M. (1974) Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society Series B,* **36** 111–147.

## Appendix Technical Details

This appendix contains some further details and technical information that allow our analysis to be reproduced.

### Bayes Rules

Under 0–1 loss the Bayes rule chooses the class with maximum posterior probability

$$c(x) = \texttt{arg}\max_{k=1,\cdots,K} p(k \mid x).$$

When $L(j,k) = \mid j - k \mid$ the Bayes rule can be writtrn as

$$c(x) = \texttt{arg}\max_{k=1,\cdots,K}\Big\{p(k \mid x) - \sum_{j\neq k}(\mid j - k \mid -1)p(j \mid x)\Big\},$$

where $p(j \mid x) = 0$ for $j \neq 1, \ldots, K$. Observe that the classifiers for these two loss functions are similar, but under $L(j,k) = \mid j - k \mid$ classes are favoured when, if chosen, they would result in a lower probability of making a serious error.

### Network Details

NOLR takes $\eta(x)$ to be the single linear output of a feed-forward neural network with input $x$, skip layer connections and sigmoid transfer functions in the hidden layer. Then for weights $w_{ij}$ and biases $\alpha_j$ we have

$$\eta(x) = \sum_{i\to o} w_{io}x_i + \sum_{j\to o} w_{jo}\sigma(\alpha_j + \sum_{i\to j} w_{ij}x_i),$$

where $\sum_{i\to j}$ denotes the sum over $i$ such that unit $i$ is connected to unit $j$, and unit $o$ is the single output unit. The usual output bias is effectively incorporated in the cut-points. Observe that OLR is the special case of NOLR with no hidden units. Hastie & Tibshirani (1987) considered using a form of $\eta(x)$ that is a linear combination of smooth functions of the components of $x$.

**Regularization**

Ripley (1994a) gives a Bayesian interpretation of weight decay as putting a prior distribution on the weights. In the NOLR model, weights on the connections to the output unit are excluded from regularization to avoid limiting the range of the cut-points. It is advantageous to regularize the weights in groups, giving different decay parameters to the set of skip-layer weights and the set of bias weights. Experimentation shows that the cut-points tend to become very widely separated when over-fitting occurs. We regularize to discourage this by imposing independent Gamma prior distributions on the differences between adjacent cut-points. This is equivalent to adding the regularization term

$$E(\boldsymbol{\phi}) = -\sum_{i=2}^{K-1} \left[ t(\phi_i - \phi_{i-1}) + (\alpha - 1)\log(\phi_i - \phi_{i-1}) \right],$$

where the mean and variance of the prior are determined by $\alpha$ and $t$, which can be chosen by cross-validation. Now we minimize $-\log \ell(\mathbf{w}, \boldsymbol{\phi}) + \lambda D(\mathbf{w}) + E(\boldsymbol{\phi})$. Regularization is less severe than fixing the cut-points, which experimentation suggests usually requires a network of considerably more units to achieve the same degree of fit. As the objective is merely to prevent very large separation of the cut-points the prior will not be particularly strong. It is reasonable to choose means such that the probability for each class dominates at its maximum.

**Comparing Classifiers**

General techniques for comparing the performance of various classifiers are discussed by McLachlan (1992) and Ripley (1994b). We give some specific details for ordered classes. When the loss is a function of the difference between the true and assigned classes we can obtain a simple expression for the standard error on the total loss. Let $M_i$ be the number of test samples where the difference between the true and predicted class is $i$ ($i = 0, \ldots, K-1$). Then the $M_i$ have a joint multinomial distribution. If the loss incurred in assigning a sample to a class distance $i$ from its true class is $l_i$, and $m_i$ is the observed value of $M_i$ then the estimated standard error on the total loss on $n$ samples is

$$s = \frac{1}{\sqrt{n}} \left( \sum_i l_i^2 m_i (n - m_i) - 2 \sum_{i>j} l_i l_j m_i m_j \right)^{1/2}.$$

This gives an approximate 95% confidence interval of $\pm 2s$. So the estimated standard error of the difference in the error rate between two classifiers with estimated standard errors $s_1, s_2$ is $(s_1^2 + s_2^2)^{1/2}$, under the assumption that the test sets are independent. However, classifier performance will almost always be compared using the *same* test set, so comparisons are more accurate than the above standard errors imply. If we treat the a comparison of loss incurred using two classifiers on the same test set as a blocked experiment, a paired $t$-test is appropriate.

**Further Details of the Equity Rating Example**

Linear, quadratic and logistic discrimination all performed significantly worse than both OLR and NOLR. $k$-Nearest neighbour techniques performed less well than all other methods even after careful metric selection. There was no gain from running six passes of the *condensing* algorithm (which removes members of the training set that are misclassified on re-substitution; Devijver & Kittler, 1982). Increasing $k$ improves performance as it is effectively using a wider-range density estimation, reflecting the high noise component apparent in the data.

For the network methods, many local optima in the fitting criteria were found. These were genuine local optima rather than early stopping by the optimizer: the Hessian matrix was positive definite. (The Hessian was calculated analytically exploiting the ideas for efficient computation of $\partial^2\eta/\partial w_{ij}\partial w_{kl}$ given by Bishop, 1992). As in the synthetic example, for each architecture and training set we average the predicted probabilities for the test set over five runs of the optimizer from randomized starting weights. The performance of this averaged classifier was often significantly better than any of the individual ones. As a caveat (rather than a motivation for 'early stopping') we note that when a less sophisticated optimizer was used, fewer local minima were found.

**Practical Model Fitting for NOLR**

We give some brief details concerning fitting the NOLR model. In general, optimization for NOLR is more difficult than for networks of the same size fitted by least squares or softmax. Fitting is complicated by the order constraints (3) on the cut-points and by the poor scaling of the network parameters $\mathbf{w}$ relative to the cut-points $\phi$. Training time is reduced by exploiting knowledge of the form of the likelihood surface whilst optimizing. For example, solutions will occur when none of the order constraints are active.

Convergence using optimization algorithms designed for linearly constrained problems tends to be extremely slow since the log-likelihood is infinite on the boundary of the feasible region. This results in a tendency for the optimizer to spend too long near the boundary. The BFGS quasi-Newton method (Gill *et al.*, 1981) performs well in maximizing the log-likelihood, having first reperameterized the cut-points as $\psi_i = S_i \exp(\phi_i - \phi_{i-1})$ to remove the order constraints and improve the scaling (by adjusting the constants $S_i$). The optimizer is supplied with the analytic first derivatives of $\log \ell$ with respect to the weights and the cut-points, and considerable care is required in updating the approximate inverse Hessian to avoid slow ascent in flat regions of the likelihood surface. Randomized starting weights and input variable ranges should be chosen to prevent saturation of the logistic inputs, so avoiding almost flat regions of the likelihood. Optimization is usually fastest when the cut-points are held fixed for the first few iterations.