Modern Pattern Recognition Methods for Ordered Classes

Ravi Kalia
Department of Statistics
University of Oxford
Oxford, 0X1 3TG
kalia@maths.ox.ac.uk

October 31, 2006

Abstract

This is a review of well-established and contemporary methods for the task of classifying patterns into ordered classes. These methods are compared for the classification task of two sets of data, artificial data and investment data from a real-world problem in financial forecasting. Suggestions are made for possible avenues of further research in ordinal regression.

1 Introduction

Models for the analysis of ordinal, or ranked, data have been developed initially by the statistics community and more recently by the machine learning community. In both paradigms, the development of models for multi-class or nominal data preceded analysis of ordinal data. This is due to the nature of the problem which is between regression and classification, but not either.

Although it is possible to use multi-class procedures for ordinal response data, it is desirable that the rank nature of responses be incorporated in the classification task. Exploiting ordinality will lead to simpler models and consistency in the derived decision rule.

At the modelling stage a decision has to made whether the ranks of classes are known. Depending on the problem it may be obvious how to rank the classes. If it is not then we may question whether ranked responses are a genuine feature. For the most part, in this report, it is assumed that the class ranks are known.

The most commonly used approach is to rely on a latent random variable, whose realisations determine the response of the ordinal variable. The ordinal response is chosen by the interval that the latent variable falls in. The goal then becomes one of estimating the distribution of the latent variable and how the regressors relate to it.

Likert (1932) proposed modelling ordinal variable by coding them as integers corresponding to an ordinal scale. One being for the least preferred class, two for the second least preferred class and so on. Statistical analysis can be performed with the re-coded data.

The mean response approach, treating the problem as one of linear regression on the ordinal categories, is discussed in Agresti (2002). This is based on models from Bhapkar (1968), Grizzle et al. (1969); Williams and Grizzle (1972).

Amongst the more prominent perspectives, models have been developed which use a similar approach to the GLM paradigm. McCullagh (1980) formulated the first models for ordinal regression. These were the proportional odds and proportional hazard models. Detailed analysis can be found in McCullagh and Nelder (1983). This was expanded on by Anderson (1984) for the type of ordinal variable and unifying a model for ordinal and nominal data. These models,taking a GLM approach, rely on a link function from the cumulative distribution function to the latent variable. The most popular link is the logistic function. Agresti (2002) illustrates the adjacent-categories logits as well as the proportional odds and proportional hazards model. The Bayesian perspective on GLMs for ordinal data is available in Albert and Johnson (1999).

A non-linear approach using neural networks to model the latent variable has been suggested by Mathieson (1995), (1998) and Da Costa and Cardoso (2005). By applying a link function to the cumulative distribution function of the ordinal variable, the latent variable is modelled through a neural network, and determines posterior class probabilities using a suitable loss function and prior class probabilities. Mathieson (1998) provided analytical approximations to the predictive class probabilities. Da Costa and Cardoso (2005) rely on a single output from the neural network and then use this in a binomial distribution to determine the posterior probabilities for each class.

Support Vector Machines (SVMs) are a new and exciting methodology for non-linear modelling. A classification method which applies linear separating methods in a non-linear, possibly infinite space, the major advantage of support vector methods is that they by-pass extensive parameter estimation of non-linear function, through a maximum margin approach (Vapnik et al. 1992, Weston 1999, Knerr et al. 1990, Herbrich et al. 1999, Shashua and Levin 2003, Crammer and Singer 2002, Chu and Keeerthi 2005, Platt 1999a

1999b, Keerthi et al. 2001). Details are discussed later in the report.

In what follows, the problem of pattern recognition for ordered classes is formally defined and placed in a decision theory framework.

1.1 Classification Task

Given a pattern $\mathbf{x}$ the task is to classify it as belonging to a class $C_{k_1}, \ldots, C_{k_K}$. The matter is complicated since there exists an order or preference relation, $>$, such that for any two different classes C_{k_i} and C_{k_j}:

1. Completeness

   either

   - $C_{k_i} > C_{k_j}$

   or

   - $C_{k_j} > C_{k_i}$

2. Transitivity

   If $C_{k_i} > C_{k_j}$ and $C_{k_j} > C_{k_l}$ then this implies $C_{k_i} > C_{k_l}$.

Both properties hold for ordered classes, therefore the order relation is a rational preference relation.

It is because of the order relation that the problem is more than just a classification, and yet because of a lack of inherent distance between classes it is not one of regression.

It is usually assumed that the preference relation and number of classes is known in advance. Determining these quantities is an avenue of research discussed later in this report.

We can assume that the ordinal classification is a realisation from stochastic variable Y with the least preferred class having label 1, the second least having label 2 and so on. In constructing the rule we assume that the data available are an independent identically distributed sample from the pattern x and an ordinal classification $y \in \{1, ..., K\}$ which has a distribution function defined by $P_{X,Y}$.

An objective function is then required which incorporates the data and assumptions about the penalty for the extent of misclassification. The optimizers of the objective function are used to define the classification rule.

Although not explicitly necessary for the classification exercise we may also be interested in some underlying probabilities. For example we may be interested in $P(y|x)$; the probability of a given pattern $\mathbf{x}$ belonging to a particular class y.

1.2 Decision Theory

In evaluating classification we need to consider the penalty for misclassification. Suppose that we classify a pattern as c(x), when it is actually of class y. Then the loss for misclassification is given by L(c(x),y). It is usual to define a classification rule which minimizes the expected loss, with respect to the posterior probability of classification given the data.

There are three popular loss functions, each being well suited to nominal classification, regression, and feasibly for ordinal regression.

- 0-1: $L\left(c(x), y\right) = 1_{\{c(x)=y\}}$
- Quadratic: $L(c(x), y) = [c(x) - y]^2$
- Absolute: $L(c(x), y) = |c(x) - y|$

These loss functions correspond respectively to the mode, mean and median class of the posterior probability of the class given the data.

The 0-1 loss function gives an equal loss for misclassification of a patten to any of the other classes. It does not penalize for a misclassification that is one class away from say one that is the maximum number of classes away. Therefore it is seen as a natural choice for nominal classification problems.

The quadratic loss function measures the square of the difference between the prediction and the actual value. The loss function naturally aligns itself for regression, incorporating both distance and ordering.

The absolute difference loss function measures the number of classes that the misclassification is away from the actual class. It will penalize for misclassification and increase the penalty by the number of classes that the misclassification is out by. It is often favoured for use in ordinal classification problems. This will be the loss function used for the models used in this report.

Having said that, although the absolute loss function suffices it is a symmetric function, so that over-estimating, is treated the same as under-estimating. There are many practical scenarios where there would be a different loss incurred for under-estimating compared to over-estimating. For example, in classifying the condition - *Imminent failure, Weak, Good, Full strength* - of a critical component used for a hazardous task, the loss from classifying a true state 'Weak' as 'Good' will be higher than classifying 'Imminent failure'. The absolute loss function cannot capture this difference.

1.3 Outline

The structure of this report is to discuss existing methodology for ordinal classification;

- Ordinal Regression
- Neural Networks
- Support Vector Machines

Where available, implementations in R for ordinal classification are considered for two datasets, one artificial the other from investment data.

Thoughts on further research avenues are considered along with conclusions.

2 Ordinal Regression

Ordinal regression was one of the earliest approaches developed to classify ordered classes. A naive approach of regressing covariates on coded class labels is the simplest model. The distance between classes is set arbitrarily to perform the regression.

Models that specifically deal with ordinal variable have relied on proportional hazards, proportional odds or adjacent cells. The approach has been to parameterize a latent variable and it's mapping to the ordinal categories. Following the GLM approach, the latent variable is treated as being a function of a linear predictor. It is calculated as a linear combination of the regressors. Extending the linear approach, non-parametric methods are analyzed. Inference of parameters using maximum likelihood and Bayesian viewpoints are considered. Non-linear techniques to model the latent variable are also discussed.

Below we detail these approaches to ordinal regression.

2.1 Models

A naive approach to modelling ordinal variables is to perform regression, either linear or non-linear (Moody and Utans 1995) on the ordinal labels, say 1 to K suitably, re-coding the original labels where needed. This ignores the problem that there is no metric between the ordered classes. It will respect the order relations, but the lack a true distance will lead to results that have no understanding of the probabilities involved and it does not guarantee any optimality in the rule chosen. Mathieson (1998) has suggested

that the integer labels can be replaced with scores and that these could also be estimated in the optimisation process.

McCullagh (1980) provides an excellent account of ordinal regression strategies. The GLM approach is modified in that the link function is applied to the CDF of the ordinal variable to relate to the latent variable.

$$LINK\left[P(Y \leq y|x)\right] = \eta(y,x) = \theta_y - \beta^{\mathbf{T}}\mathbf{x} \tag{1}$$

and the added constraint that $\infty = \theta_K \geq \theta_{K-1} \ldots \geq \theta_1 \geq \theta_0 = -\infty$ which are cut-points of the continuous scale.

We can think of the latent variable as being related to the ordinal such that

$$Y = j|\mathbf{x} \quad \text{if} \quad \theta_{y-1} < Y^*|\mathbf{x} \leq \theta_y$$

Then with $Y^* = \beta^{\mathbf{T}}\mathbf{x} + \varepsilon$, Agresti (2002), the inverse of the link function determines the distribution of ε, which we may think of as an error.

For a logistic link function

$$LINK\left[P\right] = log\left[P/(1-P)\right]$$

the distribution of the error, ε, is the logistic distribution.

The logistic link leads to the proportional odds model. The odds of $\{Y \leq j|\mathbf{x}\}$ are

$$ODDS\left[\{Y \leq j|\mathbf{x}\}\right] = \frac{P(Y \leq j|\mathbf{x})}{1 - P(Y \leq j|\mathbf{x})} = K(j,x) = K_j e^{-\beta^{\mathbf{T}}\mathbf{x}}$$

Then the odds for two different treatments with $Y \leq j$ is

$$ODDS\left[\{Y \leq j|\mathbf{x_1}\}\right] / ODDS\left[\{Y \leq j|\mathbf{x_2}\}\right] = e^{\beta^{\mathbf{T}}(\mathbf{x_2}-\mathbf{x_1})}$$

Therefore the odds are proportional to each other. Applying the logit model and after re-parameterization the model is as (1) with $\theta_j = ln\left[K_j\right]$.

Under the proportional hazards model, the survival function is related to the regressors as follows

$$S(j) = P(Y > j|x) = e^{-\Lambda(j)} = e^{-\Lambda_0(j)e^{-\beta^{\mathbf{T}}\mathbf{x}}}$$

where $\Lambda(j) = \int_0^j \lambda(j)$ and $\lambda(j) = \lambda_0(j)e^{-\beta^{\mathbf{T}}\mathbf{x}}$. It is easily verified that the hazard rate and integrated hazard functions are proportional for a given threshold and different treatments.

Applying the complementary log-log link function (c-l-l) to the CDF of the ordinal variable gives

$$log(-log(1 - P(Y \leq j|\mathbf{x}))) = \eta(y, x) = \theta_y - \beta^{\mathbf{T}}\mathbf{x} \tag{2}$$

Using this link function corresponds to the proportional hazards model. It also implies that the underlying distribution of ε is an extreme value distribution.

Using a probit link function implies that the distribution of ε is the standard normal. Other link functions can be used, however it is not clear how one should interpret the parameters.

Agresti (2002) discusses adjacent categories logits model. These are an alternative to using cumulative probability models. This model performs a regression on the logit of $P(Y = y|Y = y \quad \text{or} \quad Y = y + 1)$, which simplifies

$$logit\left(P(Y = y|Y = y \quad \text{or} \quad Y = y + 1)\right) = logit\left(\frac{P(Y = y)}{P(Y = y \quad \text{or} \quad Y = y + 1)}\right)$$

$$= log\left(\frac{\frac{P(Y=y)}{P(Y=y \quad \text{or} \quad Y=y+1)}}{1 - \frac{P(Y=y)}{P(Y=y \quad \text{or} \quad Y=y+1)}}\right) = log\left(\frac{P(Y = y)}{P(Y = y + 1)}\right)$$

$$= log\left(\frac{\pi_y(x)}{\pi_{y+1}(x)}\right) = \theta_j + \beta^{\mathbf{T}}\mathbf{x}$$

A convenient property of adjacent categories logits is that after re-parametrization they can be expressed as baseline category logits. This model can be used to identify those explanatory variables which have differing variables across adjacent categories.

Anderson (1984) advocated that the ordinal regression be split into two separate cases depending on whether there is some knowledge of the underlying scale from which the ordinal classes have been constructed. For example, for grading on a scale of (*Poor, Satisfactory, Good, Very Good, Excellent*), knowing that the scale is made from quintiles of percentage scores should be treated differently from the case when there is no knowledge of the underlying scale.

Anderson also argues for a *stereotype* model for ordinal regression. In it, the coefficients of regression are allowed to vary for different thresholds of the ordinal scale. However he constrains the coefficients of regression to parallel, so that they only differ by a scaling. The novelty of this approach is that

the scalars are used to determine the how to order the classes. Indeed, the model can decide if ordering the classes is an important feature at all.

Given that usually there are a small number of classes, potentially all orderings of the classes could be tested to decide on a best fit. That would be needed if there is some doubt about the ordering of the classes.

Mean response models where a weighted sum of the underlying probabilities is used as a response to fit a linear model have been examined by Bhapkar (1968), Grizzle et al. (1969), Williams and Grizzle (1972). However the optimization is non-trivial as determining the probabilities is treated as a separate exercise using a multi-nomial likelihood function. This model benefits from providing a simple explanation of the effects of the regressors on the category. Also as K increases the model becomes closer to ordinary regression models. However treating ordinal variables in a quantitative manner will necessarily lead to higher prediction errors than models which work with the ordinality of the data directly.

2.2 Inference

Estimating the parameters can be done using maximum likelihood and Bayesian inference. In order to do this the likelihood needs to written down explicitly in terms of the parameters.

For the cumulative logit model we define the likelihood below.

Suppose that, for an ordinal variable with K classes, we observe for each pattern $\mathbf{x_i}, \quad i = 1, ..., n$ a frequency vector of $\mathbf{y_i} = (y_{i1}, y_{i2}, \ldots, y_{iK})$.

Writing $P(Y = j|\mathbf{x}) = \pi_j(\mathbf{x})$ and using the notation above, the likelihood for this sample is then

$$\begin{aligned} L &= \prod_{i=1}^{n}\prod_{j=1}^{K} \pi_j\left(\mathbf{x_i}\right)^{y_{ij}} \\ &= \prod_{i=1}^{n}\prod_{j=1}^{K} \left(P(Y \leq j|\mathbf{x_i}) - P(Y \leq j-1|\mathbf{x_i})\right)^{y_{ij}} \\ &= \prod_{i=1}^{n}\prod_{j=1}^{K} \left(\frac{\exp(\theta_j + \beta^{\mathbf{T}}\mathbf{x_i})}{1+\exp(\theta_j + \beta^{\mathbf{T}}\mathbf{x_i})} - \frac{\exp(\theta_{j-1} + \beta^{\mathbf{T}}\mathbf{x_i})}{1+\exp(\theta_{j-1} + \beta^{\mathbf{T}}\mathbf{x_i})}\right)^{y_{ij}} \end{aligned} \tag{3}$$

The maximum likelihood estimator of the parameter will not be in closed form, but can be found using numerical algorithms.

Taking logarithms of the functions, simplifies the objective function while exploiting the monotonic property of the logarithmic operator. Although

there are a number of optimization algorithms, the optimizers can be found using Fisher scoring for multivariate GLMs, which is discussed in McCullagh (1980), Thompson and Baker (1981) and Walker and Duncan (1967). It has been shown that the log-likelihood is concave and well behaved for many link functions, so that there is a single optimum which numerical methods converge to rapidly.

Taking a Bayesian perspective, the above models can be viewed as having uncertain regression parameters. The specification of a prior distribution is desirable if their are reasons to believe that particular parameter values are more likely than others. The quantity of interest is the predictive distribution of the class given a pattern. We average the estimate of class probabilities over the parameter space, using the prior to weight the class probabilities. This produces final estimates that are smaller than the MLE point estimate. However, it is often difficult to evaluate analytically the integrals to obtain the predictive distributions. In the past using series expansions to approximate the integral provided a crude solution. More recently using Markov Chain Monte Carlo (MCMC) methods, Albert and Johnson (1999), to approximate integrals has become feasible using commonly available computing resources.

2.3 Extensions

Hastie and Tibshirani (1987) extended the ordinal regression beyond the GLM paradigm with the additive functions. Rather than use a model with a linear relationship being fitted to the logit transform of the CDF of an ordinal variable

$$logit\left[P(Y \leq y|x)\right] = \theta_y - \beta^{\mathbf{T}}\mathbf{x}$$

they used additive non-parametric functions to model the relationship between the logit transform and the covariates.

$$logit\left[P(Y \leq y|x)\right] = \theta_y - \sum_{j=1}^{j=K} f_j\left(x_j\right), \quad E\left[f_j\left(x_j\right)\right] = 0 \quad \forall j$$

where the functions f_j are estimated using scatterplot smoothers, although other non-parametric estimation procedures could be used also. The approach is to fit the functions f_j locally so as to provide better predictive accuracy. One of the problems then becomes one of interpret-ability of the model.

Although referred to as non-parametric functions there are usually many tuning parameters which have to be chosen to make the model fit well.

2.4 Neural Networks

Neural networks offer a flexible approach to infer a function from observations. The approach is inspired by the operation of the animal brain. The function is constructed additively from other functions which them-self may be constructed from other functions additively. Going back far enough the process relies on the input data or pattern. The weights taking each function to the next are determined as the optimisers of a loss function which scores the accuracy of the neural network. Good introductions to neural networks are to found in Bishop (1995) and Ripley (1996).

Neural networks have been studied extensively for classification and regression problems. The use of neural networks for ordinal classification has not been studied as intensively.

Mathieson (1995) uses the approach of modelling the latent variable underlying an ordinal variable using a neural network. He investigated the logit transform to regress cut points θ_y and a non-linear function $\psi(\mathbf{x})$, that is

$$logit\left[P(Y \leq y|\mathbf{x}))\right] = \theta_y - \psi(\mathbf{x})$$

Mathieson (1995) found that for a suitable number of hidden units, neural networks outperformed ordinal regression using a linear predictor using the plug-in maximum likelihood estimator.

In Mathieson (1998), he approximated the predictive distribution classification given a pattern using series approximation to integrate out the parameters weighted by the prior probability of the parameters. Performing MCMC was not possible computationally due to the multi-modal nature of the likelihood.

Da Costa and Cardoso (2005) have developed a novel approach to apply neural networks to ordinal regression. Instead of relying on cut-points on a continuous scale to classify ordinal variables they use the binomial distribution with a single probability parameter.

The inputs are run through a neural network to produce a single probability parameter, $p(\mathbf{x})$. This is then used to calculate the final values for K outputs, which are the probabilities that a particular input pattern belongs to that ordinal classification.

The probability that a pattern $\mathbf{x}$, belongs to a class y is given by

$$P(Y = y|\mathbf{x}) = \frac{K!}{y!\,(K-y)!} p(\mathbf{x})^y \left(1 - p(\mathbf{x})\right)^{K-y}$$

Thus the ordinal classification variable follows a binomial distribution. It benefits from the fact that by construction there will be only one mode;

at worst two which will be contiguous so that we have exactly the same probability for two classes.

In essence the model is a single output neural network whose output is then processed to reach the final outputs. The training uses a 0-1 loss function which penalizes mis-allocation according to the maximum posterior probability criteria.

Finally, we review literature on ordinal classification using support vector machines.

3 Support Vector Machines

Support vector machines have recently attracted much attention because they demonstrate as great, and sometimes better generalisations than other machine learning methods. Good introductions to the methodology are available in Burges (1998); Duda, Hart and Stork (2001); Cristianini and Shawe-Taylor (2000).

We explain how Support Vector Machines have been developed for the task of ordinal classification.

3.1 Binary classification of linearly separable patterns

The simplest problem, from which support vector machines are developed is binary classification of linearly separable data. Observations are collected such that

$$\{\mathbf{x_i}, y_i\}_{i=1}^{n}. \quad \text{where} \quad \mathbf{x_i} \in \Re^p \quad \text{and} \quad y_i \in \{-1, 1\}.$$

For now we assume that the data are separable using a hyperplane. We then want to find the best hyperplane. Since the task is to classify existing patterns and generalize for unseen patterns, a plane which is the maximum distance from the nearest patterns of either class. Thus the problem does not need all the data, just the convex hull - or support vectors - of the patterns for each class.

Let the hyperplane be defined by the equation

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = \mathbf{0}$$

So we wish to determine $\mathbf{w}$ subject to maximising the distance between the support vectors and the plane. Using vector calculus and Lagrange constraints, the optimization problem is then

$$L_p = \frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{p} \alpha_i y_i \left(\mathbf{w}\mathbf{x_i} + \mathbf{b}\right) + \sum_{i=1}^{p} \alpha_i$$

We use the notation L_p as this is the optimisation function for the primal problem, whose dual is actually solved to find the optimizers. When this function is minimised with respect to $\mathbf{w}, \mathbf{b}$ and the Lagrange multipliers α_i subject to the constraint that $y_i \left(\mathbf{w}\mathbf{x_i} + \mathbf{b}\right) > 0, \quad \alpha_i > 0$ while the derivative of L_p with respect to the α_i must vanish. The optimizers determine the hyperplane. That is $\mathbf{w_{opt}}, \alpha_{\mathbf{opt}}$ such that

$$L_p(\mathbf{w_{opt}}, \alpha_{opt}) \leq L_p(\mathbf{w}, \alpha) \quad \forall \quad \mathbf{w}, \alpha : L_p^{'}(\alpha) = \mathbf{0}$$

This problem is one of quadratic programming for which well defined methods exist - faster optimisation algorithms will be discussed later. Due to the geometry of the surface a single optima is guaranteed. For convenience we understand w to be the optimizer w_{opt} outside of the optimization.

The result is a decision rule which allocates between classes $\{-1, 1\}$ according to

$$c\left(\mathbf{x_i}\right) = sign(\mathbf{w} \cdot \mathbf{x_i} + \mathbf{b})$$

This is a simplified case. However the practicalities below need to be dealt so that we have a useful method;

- Non-linearly separable data
- Misclassifications in the training set
- Fast calibration of parameters
- Multi-class classification

Of course of paramount interest to us, applying these methods to ordinal classification. All of these issues will be discussed in building up to the use of support vector machines for ordered data.

3.2 Binary Kernel Trick

As mentioned above, the technique needs the data to be linearly separable by class. This restricts the usefulness. Boser, Guyon and Vapnik (1992) proposed using the 'Kernel Trick'; a way to adapt linear methods to fit non-linear data.

In order to do this we first need to note that the optimisation problem can be restated to its dual. The optimisation problem is now to maximise L_p with respect to α while the derivatives of L_p with respect to $\mathbf{w}, \mathbf{b}$. That is $\mathbf{w_{opt}}, \alpha_{\mathbf{opt}}$ such that

$$L_p(\mathbf{w_{opt}}, \alpha_{opt}) \geq L_p(\mathbf{w}, \alpha) \quad \forall \quad \mathbf{w}, \alpha : L_p^{'}(\mathbf{w}, \mathbf{b}) = \mathbf{0}$$

This is called the Wolfe dual of the problem. The optimizers are the same as the original problem. This representation allows for the kernel trick to be applied.

The kernel trick is to transform the data usually to a higher dimensional, possibly infinite space, such that even though x maps non-linearly to the ordinal response, by applying a vector function ϕ to the pattern x - usually higher dimensional - sometimes infinite dimensional space. For this

$$\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_M(\mathbf{x}))$$

can be modelled as linearly separable by class.

Due to Mercer's condition, these transformations can be characterized uniquely through a dot product.

For a function, K, which obeys

$$\int \int K(x)K(y)g(x)g(y)dxdy \geq 0 \quad \text{for all square-integrable functions K}$$

the following is true:

$$K(x) = \phi(\mathbf{x})^T \cdot \phi(\mathbf{x})$$

Often $K(\cdot)$ is referred to as a Kernel. Because of this, optimization problems with dot products in the objective function can be replaced by a suitable kernel. As a result a non-linear problem in a pattern space can be solved using linear methods in the feature space induced by the Kernel function. It is therefore not necessary to know what the non-linear functions $\phi_i(\mathbf{x})$ are. Therefore our Lagrangian can be used as before, just replacing the dot product by a kernel function.

The advantage of using the kernel trick for support vector machines is that the non-linear functions don not have to be estimated. Also the optimisation is not complicated since there is a single optima to search for.

However, choosing the kernel and its parameters is done without a theoretical grounding. Duan et al.(2001) call these variables hyper-parameters and

note that cross validation and leave one out calibration exercises are commonly used. They argue that to reduce computational cost, crude analytical approximations should be used. A simple suggestion has been to perform a grid search over the parameters (Meyer 2006) until the best generalisation performance is achieved.

3.3 Binary misclassification in the training set

Because of the ability of support vector machines to separate out classes non-linearly, we need to wary of noisy data significantly undermining the true nature of the patterns in the training set.

Suppose that some of the data in the training set has been mislabelled. Then it would be prudent to penalise the contribution of those observations that are outliers relative to the rest of the sample. The mathematical formalism for this is introduced via a slack variable ξ_i. We now require that the the hyperplane $\mathbf{w} \cdot \mathbf{x_i} + \mathbf{b} = \mathbf{0}$ is such that the following hold

$$\mathbf{w} \cdot \mathbf{x_i} + \mathbf{b} \geq \mathbf{1} - \xi_\mathbf{i}$$

$$\mathbf{w} \cdot \mathbf{x_i} + \mathbf{b} \geq -\mathbf{1} + \xi_\mathbf{i}$$

subject to $\xi_i \geq 0, \quad \forall \quad i$

Ideally we wish to minimise the slack variables as much as possible. We also associate a cost with this slack variable on the objective function (Cortes and Vapnik 1995).

$$L_D = \sum_i \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j$$

As before the dot product can be replaced by a kernel function. The problem is still a quadratic programming one with unique optima, only now we are factoring misclassification explicitly into the model. Finding the optimizers can be time consuming using standard quadratic programming problems. Specialised methods have been developed to lower the computational time to perform the optimisation as the size of the data increases.

3.4 Binary optimization

As seen above in order to extract the decision rule is is necessary to solve a quadratic programming problem. Vapnik (1982) showed that the quadratic programming problem can be broken down into smaller quadratic programming problems by removing examples that have a Lagrange multiplier that

is zero. These smaller problems can then be solved using numerical methods. Details on how to reduce the time taken to optimize can be found in Platt (1999a) and Keerthi et al. (2001).

Next we look at how this binary classification algorithm can be modified for the multi-classification task.

4 Multi-classification task

Several models have been to make SVMs perform classification for multiple classes. The unifying theme is to perform binary SVM redefining classes and then combine the results into a classification rule.

Duan and Keerthi (2005) investigated three popular methods in tests against benchmark datasets.

- One against All

  Under this scheme each class is fitted against all others in the training set with a support vector machine. Then for each new pattern an output is associated with every class. The class with the best output is the label for the pattern (Hsu and Lin 2002).

- One against One

  More support vector machines are needed using this method. Each pair of classes have a support vector machine fitted. Then each new pattern is tested on each of the $\frac{K(K-1)}{2}$. The class that has collected the most classification decision votes is where the pattern is assigned to.

- Pairwise Coupling

  Assuming that the outputs of support vector machines can be treated as posterior probabilities Hastie and Tibshirani (1998) constructed an algorithm to couple the probabilities and estimate the posterior probability of each class. The class with the highest probability is where each pattern is allocated to.

Benchmark tests have shown that each method has merit depending on the nature of the data. For example when training data is sparse the Pairwise coupling approach gives a higher classification accuracy (Duan and Keerthi 2005).

There are other methods which can be used to adapt SVMs to multi-class problems. Those mentioned above are some of the most popular implementations for computer algorithms are readily available.

4.1 SVMs for ordinal data

Having built an understanding Support Vector Machines, we now turn to our looking at how a SVM can be used for ordinal classification. The development of SVM methods to deal with ordinal classification has only recently received attention in the literature. All rely on modelling a latent variable and cut-points which depend on the pattern $\mathbf{x}$. The assumption is that realisations of the latent variable are allocated to bins defined by the cut-points and this determines the value that the ordinal variable takes.

Herbrich et al. (2000) have a theoretical model for the empirical risk involved with ordinal classification and provide an upper bound for this in terms of a distribution independent risk functional. They begin with the assumption that classification rule must be asymmetric and transitive. They enforce this by arguing for the existence of a linear utility function. A mapping from a partition of intervals from the range of the utility function to the ranks in the data yields the classification rule.

The optimisation procedure is modified to incorporate these features of the model. It is still a quadratic programming problem and the estimation of the cut-points is carried out after the weight parameters have been estimated. Herbrich et al. report that their algorithm outperforms using multi-classification SVM methods for ranking information retrieval data.

Crammer and Singer (2002) work with parallel hyperplanes. Their idea is to have a parallel hyperplanes with differing intercepts. They estimate the parameters by minimising a loss function iteratively. The decision rule classifies patterns to ranks depending on which hyperplane is closest to the pattern. They also have an online approach, so that as the true rank of the pattern is revealed the parameters of the decision rule are updated if there has been a misclassification. They note that it is possible to combine this approach with the kernel trick to model non-linear data.

Shashua and Levin (2003) introduce two approaches to determine the optimal hyperplanes under differing optimisation criteria. Under the first, they maximise the distance between each pair of consecutive ranks. Under the second criteria the sum of squared distances between hyperplanes is maximised. For both cases the optimisation problem is one of quadratic programming which has a unique optima.

Chu and Keerthi (2005) demonstrate that Shashua and Levin failed to implement a constraint in their approach, which is not consistent with the ranking of the data. They offer a way to implement the constraint and improve the time it takes to run the optimisation by modifying the SMO algorithm. They also suggest a new approach that maximizes the distance between the hyperplanes by using training data from all ranks. Their algorithm does not require the order constraint to be made explicit.

Now we investigate how well some of the models reviewed perform at generalization from the training set.

5 Empirical Work

The work in this section is from preliminary results which have not yet been verified. Their use was intended to illustrate that there are situations where non-linear methods such as SVMs outperform linear methodologies. However, at this stage it is not clear what the fabricated example and investment data actually demonstrate. Nonetheless, we can see where the work is heading for now.

Here we compare how proportional odds logistic regression, probit regression, complementary log-log, cauchit and Support vector machine perform. Two datasets are considered. The first artificially generated data and the second data from a financial forecasting problem. In the first there are five ordered classes and in the second there are three.

5.1 Artificially Generated Data

How the data is generated from a bi-linear function used by Herbrich et al (1999). Their objective was to show that using lineaer ordinal regression strategies performed badly for this example. They found that SVMs produced superior results for this example. The function and how the data are formed is discussed below.

Given two inputs (x_1, x_2) a deterministic function and some noise map to a scalar function. This function is latent and the class orders are related through cut-points on the real line. The bi-linear function and mapping are such that:

$$y = i \Leftrightarrow (10((x_1 - 0.5) \cdot (x_2 - 0.5)) + \epsilon) \in (\theta(r_{i-1}, \theta(r_i))$$

where ϵ is normally distributed with zero mean and unit variance, and $\theta(\cdot)$ are the cut-points.

We simulate 1000 pairs (x_1, x_2) uniformly from the unit square and then calculate the corresponding ordinal classification. The next stage is to fit models which can predict the ordinal pattern as accurately as possible by training the model with some data and then comparing the performance against test data. For this purpose we randomly split the data into 2/3 for training and 1/3 for testing the generalization performance of the model. The performance is analyzed by looking at tables of predicted classes versus actual classes for the test dataset.

At this stage my results do not correspond to those of Herbrich et al. (1999). The results do however show that SVMs perform better than linear ordinal regression strategies.

5.1.1 Proportional Odds Logistic Regression

Training the model to optimise the likelihood, we can then classify the test data by classifying each pattern to the class that has the highest predictive probability.

A cross table of the predicted to true class of the test data illustrates the performance of the model.

	true				
pred	1	2	3	4	5
1	0	0	0	0	0
2	55	71	29	70	54
3	0	0	0	0	0
4	0	0	0	0	0
5	18	13	2	10	11

Similar tables are constructed for each of the methods below. In each case the model is trained and the its generalisation performance assessed with the test data.

5.1.2 Probit Regression

	true				
pred	1	2	3	4	5
1	11	11	8	9	12
2	50	47	10	46	37
3	0	0	0	0	0
4	12	26	13	25	16
5	0	0	0	0	0

5.1.3 Complementary Log-Log Regression

	true				
pred	1	2	3	4	5
1	16	19	11	21	18
2	50	50	13	46	38
3	0	0	0	0	0
4	7	15	7	13	9
5	0	0	0	0	0

5.1.4 Cauchy Latent Variable Regression

	true				
pred	1	2	3	4	5
1	0	0	0	0	0
2	55	56	21	50	41
3	0	0	0	0	0
4	18	28	10	30	24
5	0	0	0	0	0

5.1.5 SVM

	true				
pred	1	2	3	4	5
1	34	28	2	8	1
2	20	33	14	37	17
3	0	0	0	0	0
4	1	12	6	17	10
5	1	12	15	29	36

5.2 Investment Data

The data modelled here is used for financial forecasting. The dataset provides company specific metrics - accounting ratios, personnel numbers, sector performance, etc. We wish to extract a a pattern to predict stock performance based on the known company statistics. A recommendation is registered on an ordinal scale set by an equity analyst as an opinion to Buy, Hold or Sell the asset.

The data has been kindly provided by Bear, Stearns International. It has been selected from the property construction sector, with the sample of size 200 hand picked by an equity associate. After training, the predicted to true profitability rating is compared for each model, as in the case of the artificial model.

5.2.1 Proportional Odds Logistic Regression

	true		
pred	Buy	Hold	Sell
Buy	35	45	3
Hold	10	47	12
Sell	9	2	37

5.2.2 Probit Regression

	true		
pred	Buy	Hold	Sell
Buy	40	80	9
Hold	14	10	13
Sell	0	4	30

5.2.3 Complementary Log-Log Regression

	true		
pred	Buy	Hold	Sell
Buy	44	30	2
Hold	10	40	20
Sell	0	24	30

5.2.4 Cauchy Latent Variable Regression

	true		
pred	Buy	Hold	Sell
Buy	21	40	8
Hold	19	50	40
Sell	14	4	4

5.2.5 SVM

	true		
pred	Buy	Hold	Sell
Buy	40	60	0
Hold	10	20	25
Sell	4	14	27

6 Further Research

6.1 Hyper-Spheres

Recall how the paradigm for SVMs is to non-linearly transform patterns to a high dimensional feature space, where linear methods can be applied.

However the transformation to a linear space is arbitrary and for suitable kernel and parameters it should be possible to approximate any space. Suppose we wish to model the data so that patterns from the same ordinal response lie in hyper-doughnuts being centred about an origin. Then nested

spheres centred at a suitable origin will define the region of each of the ordered categories. This benefits from a natural latent distance variable. Using support vectors we would want to define the radius of each hypersphere so that it maximised the distance between patterns from adjacent classes.

Intuitively, given that each hypersphere can be defined by a single radius parameter - the origin being fixed - the onus would be on the kernel transformation to provide good separation. This may have implications for the time taken to calibrate the model.

6.2 Parallel Hyperplanes

Our thought here is to tinker with post processing once the support vector machine has been defined. Given the hyperplanes of a SVM, we wish to rotate and translate them so that the planes become parallel, the distance to each plane increasing with ordering of the patterns that are separated by it. How to rescale the distance between planes is not obvious.

Some of the existing approaches to ordinal SVMs discussed model parallel hyperplanes in the optimisation of the SVM, rather than as a post-processing exercise.

6.3 Support Vector Regression on latent variable

This approach is very much based on Mathieson (1995, 1998). For a suitable link function, say the ordinal logit transformation, we can model a latent variable with cut-points using Support Vector Regression - SVR.

Scholkopf and Smola (2001) provide a tutorial for SVR. The idea is that after a suitable kernel has been set, a linear methods can be used to fit a line of the response to the variables in the feature space. It is defined by minimizing any errors in excess on some pre-defined threshold. In order that the parameters do not over fit a penalty is added to the objective function; the sum of squares of the parameters. This makes the line tend to a flatter gradient.

I find this an appealing way to deal with ordinal data. This is because we can use support vector methods to model a latent variable which explicitly maps to the likelihood of each class. It may provide a refinement on the neural network approach of Mathieson (1995), because there is an in-built penalty for over-fitting the data. At the same time by sung the kernel trick we can break free from generalised linear models world of McCullagh (1980).

6.4 Determining the order of classes

There has not been much work on determining the ordering of classes other than the work of Anderson (1984). This could be an interesting area for investigation, particularly if the number of classes becomes large and it is not computationally efficient to test the performance of each permutation. The time taken to test all permutations scales exponentially with the number of classes for fitting each SVM. This may become an issue if the number of classes becomes substantially large or the time taken to fit each SVM is a significant cost.

We need a criteria to assess the models against each other. An obvious criteria would be the generalization performance of each model on the test set. Depending on the loss function, this may just be the percentage of mis-classifications.

A number of ideas come to mind on how to solve this problem:

- An approximation to the generalization performance of similar permutations
- Exploiting symmetry of non-linear functions in kernel trick to remove some of the SVMs
- Forward/Backwards nested model selection approach
- Random selection of permutations
- Explicit model that orders the classes. Need scalar function ϕ to score each class then.

$$\phi(Class1) < \phi(Class2) < \ldots < \phi(ClassK)$$

- Run a SVM as a multi-classification task. Then the distance - in some metric - between pairs of parameter vectors gives ranking. Else we might consider the score function is some other function of parameters and this determines ordering.

6.5 Determining whether order is relevant

There are likely to be circumstances in which it is unclear if modelling the order of classes is actually warranted. Again, there is limited literature about how this should be identified.

6.6 Rework multi-class classification

Perhaps as an intermediate step to constructing a model for ordinal classification, we could find an elegant approach to model multi-class classification using Support Vector Machines. Although methods exist and have been discussed, they lack the natural appeal of its binary classification counterpart. A different algorithm could also have extensions to ordinal classification.

6.7 Determining Kernel and hyper-parameters

For all the appeal of SVMs, the process of determining a kernel and parameters for the optimisation lacks a theoretical foundation. A systematic method based on a theoretical framework will have advantages in terms of reduced computational time against grid based searches or trial and error.

7 Acknowledgements

I gratefully acknowledge discussions of this work with Professor B.D. Ripley. I would also like to thank Bear Stearns, International Ltd research team (London, New York) for providing investment data and practical applications for the methods discussed in this report. I am indebted to Mr Alex Kuznetsov (New York) of Bear Stearns, International for his suggestions on variable selection. Further thanks are due to Dr Jeremy Stanley (New York) and Dr Sergio Pezzulli (London) of Ernst and Young LLP for discussions relating to kernel methods and machine learning strategies. I would like to thank Rajiv Menjoge of the Operations Research Centre at the Massachusetts Institute of Technology for acting as peer reviewer for this report. This research was funded by the UK Engineering and Physical Sciences Research Council.

References

[1] Agresti, A. (2002). Categorical Data Analysis. Wiley.

[2] Anderson, J. (1984). Regression and ordered categorical variables (with discussion). Journal of the Royal Statistical Society, Series B46, 1-30.

[3] Bhapkar, V.P. (1968). On the analysis of contingency tables with a quantitative response. Biometrics 24, 329-338.

[4] Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford University Press.

[5] Boser, B., Guyon, I., and Vapnik, V. (1992) An training algorithm for optimal margin classifiers. In Fifth Annual Workshop on Computational Learning Theory, 144-152. Pittsburgh, ACM.

[6] Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. Kluwer Academic Publishers, Boston.

[7] Chu, W. and Keerthi, S. S. (2005). New approaches to support vector ordinal regression. Technical report, Yahoo! Research Labs.

$http://www.keerthis.com/ord-icml-chu-05.pdf$

[8] Cohen, W. W., Schapire, R. E. and Singer, Y. (1999). Learning to order things. Journal of artificial intelligence research, 10, 243-270.

[9] Cortes, C. and Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273-297.

[10] Crammer K. and Singer Y. (2002). Pranking with ranking. Advances in Neural Information Processing Systems 14, 641-647. Cambridge, MA: MIT Press.

[11] Cristiani, N. and Shawe-Taylor, J. (2000). Suport Vector Machines. Cambridge University Press.

[12] Crouchley, R. (1995). A Random Effects Model for Ordered-Categorical Data. Journal of the American Statistical Association, Vol. 90 No. 430, 489-498.

[13] Duan, K.S. and Keerthi, S.S. (2005). Which is the best SVM Method? An Empirical Study. Submitted to Advances in Neural Information Processing Systems.

[14] Duan, K., Keerthi, S.S. and Poo, A.N. (2001). Evaluation of simple performance measures for tuning SVM hyperparameters.The 8th International Conference on Neural Information Processing.

[15] Duda, R., Hart, P., and Stork, D. (2001). Pattern Classification. John Wiley and Sons, Inc.

[16] Frank, E. and Hall, M. (2001). A simple approach to ordinal classification. In Proceedings of the European Conference on Machine Learning, 145-165.

[17] Grizzle, J.E., Starmar C.F. and Koch G.G. (1969). Analysis of categorical data with linear models. Biometrics 25, 489-504.

[18] Har-Peled, S., Roth, D., and Zimak, D. (2002). Constraint classification for multiclass classification and ranking. In NIPS-15; The 2001 Conference on Advances in Neural Information Processing Systems. MIT Press.

[19] Herbrich, R., Graepel, T. and Obermayer, K. (1999). Regression models for ordinal data: A machine learning approach. Technical report, TU Berlin. TR-99/03.

[20] Herbrich, R., Graepel, T. and Obermayer, K. (2000). Large margin rank boundaries for ordinal regression. In Advances in Large Margin Classifiers, 115-132. MIT Press.

[21] Hsu, C.-W., and Lin, C.-J. (2002) A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks, Vol. 13, 415-425.

[22] Johnson, V.E. and Albert, J.H. (1999). Ordinal data modelling (statistics for social science and public policy). Springer-Verlag.

[23] Keerthi, S. S., Shevade, S. K., Bhattacharyya, C. and Murthy, K. R. K. (2001). Improvements to Platts SMO algorithm for SVM classifier design. Neural Computation, 13: 637-649, March 2001.

[24] Knerr, S. Personnaz, L. and Dreyfus, G. (1990). Single-layer learning revisited: a stepwise procedure for building and training a neural network. In neuro-computing: Algorithms, Architectures and Applications, eds F.Fogelman Soulie and J. Herault. Berlin: Springer Verlag.

[25] Kramer, S., Widmer, G., Pfahringer, B. and DeGroeve, M. (2001) . Prediction of ordinal classes using regression trees. Fundamenta Informaticae, 47:113.

[26] Likert, R. (1932). A Technique for the Measurement of Attitudes. Archives of Psychology 140, 55.

[27] 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).

[28] Mathieson, M. J. (1998) Ordinal Models and Predictive Methods in Pattern. Recognition. D.Phil. dissertation, University of Oxford.

[29] McCullagh, P. (1980). Regression models for ordinal data. Journal of the Royal Statistical Society B, 42 (2): 109-142.

[30] McCullagh, P. and Nelder, J. A. (1983). Generalized Linear Models. Chapman and Hall, London.

[31] Meyer, D. (2006). Support Vector Machines. R interface guide.

$http://wi.wu-wien.ac.at/home/meyer/$

[32] 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, pp. 277-300.

[33] Platt, J.C. (1999a). Fast Training of Support Vector Machines using Sequential Minimal Optimization. In Advances in Kernel Methods - Support Vector Learning, B. Schlkopf, C. Burges, and A. Smola, eds., 185 - 208. MIT Press.

[34] Platt, J.C. (1999b). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In Advances in Large Margin Classifiers, A. Smola, P. Bartlett, B. Schlkopf, D. Schuurmans, eds., 61-74. MIT Press.

[35] Ripley, B. D. (1996). Pattern Recognition and Neural Networks. Cambridge.

[36] Scholkopf, B. and Smola, A. J. (2001). Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond. Adaptive Computation and Machine Learning. MIT Press.

[37] Shashua, A. and Levin, A. (2003). Ranking with large margin principle: two approaches. In S. Thrun S. Becker and K. Obermayer, editors, Advances in Neural Information Processing Systems 15, pages 937 - 944. MIT Press.

[38] Tutz, G. (1991). Choice of smoothing parameters for direct kernels in discrimination. Biometrical Journal 33, 519 - 527.

[39] Tutz, G. (2003).Generalized semiparametrically structured ordinal models. Biometrics, 59: 263 - 273.

[40] Vapnik, V. (1982). Estimations of dependences based on statistical data. Springer.

[41] Weston, J. (1999). Leave-One-Out Support Vector Machines. International Joint Conference on Artificial Intelligence.

[42] Williams, O.D. and Grizzle, J.E. (1972). Analysis of contingency tables having ordered response categories. Journal of the American Statistical Association, Volume 67 No. 337, pp. 55–63.