Generalization Bounds for Some Ordinal Regression Algorithms

Shivani Agarwal

Massachusetts Institute of Technology, Cambridge MA 02139, USA
shivani@mit.edu

Abstract. The problem of ordinal regression, in which the goal is to learn a rule to predict labels from a discrete but ordered set, has gained considerable attention in machine learning in recent years. We study generalization properties of algorithms for this problem. We start with the most basic algorithms that work by learning a real-valued function in a regression framework and then rounding off a predicted real value to the closest discrete label; our most basic bounds for such algorithms are derived by relating the ordinal regression error of the resulting prediction rule to the regression error of the learned real-valued function. We end with a margin-based bound for the state-of-the-art ordinal regression algorithm of Chu & Keerthi (2007).

1 Introduction

In addition to the classical problems of classification and regression, several new types of learning problems have emerged in recent years. Among these is the problem of ordinal regression, in which the goal is to learn a rule to predict labels of an ordinal scale, *i.e.*, labels from a discrete but ordered set. Ordinal regression is common in the social sciences where surveys frequently ask users to rate items on an ordinal scale, and has been studied previously in the statistical literature [1]. Recent years have seen a surge of interest in ordinal regression from a machine learning perspective [2–12], partly due to the fact that it is a unique problem which shares characteristics of many other learning problems such as classification, regression, and ranking – and yet is distinct from each – but also due to the fact that ordinal regression increasingly finds applications in diverse areas such as finance, medicine, information retrieval, and user-preference modeling.

Although there has been considerable progress in developing learning algorithms for ordinal regression in the last few years, in most cases, not much is known about the theoretical properties of these algorithms: how well they generalize, and at what rate (if at all) they converge to an optimal solution. In this paper, we begin an attempt to fill this gap. Our focus is on the question of generalization properties of these algorithms.

1.1 Previous Results

In the ordinal regression problem, described in greater detail in Section 2, the learner is given a sequence of labeled training examples $S = ((x_1, y_1), \ldots, (x_m, y_m))$, the x_i being instances in some instance space X and the y_i being labels in a discrete, ordered

set, which we take to be $[r] = \{1, \ldots, r\}$ for some $r \in \mathbb{N}$, and the goal is to learn a rule $g : X \to [r]$ that predicts accurately labels of future instances. The penalty incurred for a wrong prediction is larger for predictions farther from the true label: in the setting we consider, the penalty incurred by g on an example (x, y) is proportional to $|g(x) - y|$.

Barring some work on neural network models in the 1990s [13] (which was inspired largely by the statistical models of [1]), among the earliest studies of ordinal regression in machine learning was that of Herbrich et al. [2], in which a large-margin algorithm similar to support vector machines (SVMs) was proposed. This work included a margin-based generalization bound for the zero-training-error case[1]. However, the setting of [2] differs from the setting described above, in that the error of a prediction rule is measured in terms of pairs of examples for which the relative order between the predicted labels differs from the relative order between the true labels; indeed, in their setting, it is possible for a rule that predicts the wrong labels on two instances to incur no loss at all, as long as the relative order of those labels is correct. In this sense, the problem studied in [2] is more similar to some ranking problems than what has now come to be commonly accepted as the problem of ordinal regression.

The years following [2] saw several developments in ordinal regression. Crammer & Singer [14] proposed an algorithm for ordinal regression in the online learning model, motivated by the perceptron algorithm for classification, and provided a mistake bound for their algorithm. This was followed by an extension of their algorithm by Harrington [7], in which an online approximation to the Bayes point was sought, as well as extensions in [5] which included a multiplicative update algorithm. All of these came with mistake bounds; indeed, it can safely be said that these online algorithms for ordinal regression are among the better understood theoretically.

In the traditional offline (or 'batch') learning model, there have been four broad approaches to developing ordinal regression algorithms. The first approach treats the labels y_i as real values, uses a standard regression algorithm to learn a real-valued function $f : X \to \mathbb{R}$, and then predicts the label of a new instance x by rounding the predicted real value $f(x)$ to the closest discrete label. This approach can be used with any regression learning algorithm, and was discussed specifically in the context of regression trees by Kramer et al. [3]; Kramer et al. also discussed some simple methods to modify the regression tree learning algorithm to directly predict labels for ordinal regression.

The second approach consists of reducing an ordinal regression problem to one or more binary classification problems, which can then be solved using a standard binary classification algorithm. For example, Frank & Hall [4] proposed a method for reducing an r-label ordinal regression problem to a series of $r - 1$ binary classification problems, each of which could be solved using any classifier capable of producing probability estimates; the method was somewhat ad-hoc as it required certain independence assumptions in order to compute probabilities needed for making label predictions. More recently, Cardoso & da Costa [11] have proposed an algorithm for transforming an ordinal regression problem in a Euclidean space into a single binary classification problem in a higher-dimensional space.

[1] We note that the bound in [2] contains a mistake, although the mistake is easily corrected: the article incorrectly claims that a sample of m independent instances gives $m - 1$ independent pairs of instances; this can be corrected by replacing $m - 1$ in the bound with $m/2$.

In the third approach, algorithms are designed to directly learn prediction rules for ordinal regression; as in the case of [2] or [14], this usually consists of learning a real-valued function $f : X \to \mathbb{R}$ together with a set of thresholds $b^1 \leq \ldots \leq b^{r-1} \leq b^r = \infty$, the resulting prediction rule being given by $g(x) = \min_{j \in \{1,\ldots,r\}} \{j : f(x) < b^j\}$. Going back full circle to the large-margin framework used in [2], Shashua & Levin [6] proposed two large-margin algorithms, also motivated by SVMs, to directly learn prediction rules for ordinal regression; unlike [2], the problem setting in this case corresponds to the setting described above, in which the error of a prediction rule is measured in terms of the difference between the true and predicted labels. However, as pointed out by Chu & Keerthi [10], one of the algorithms in [6] contains a fundamental flaw in that it fails to guarantee the necessary inequalities among the thresholds; Chu & Keerthi offer a modification that corrects this, as well as a second large-margin algorithm that is among the current state of the art.

Finally, there has also been some work on Bayesian learning algorithms for ordinal regression, such as that by Chu & Ghahramani [8].

Among all the (offline) algorithms discussed above, only two – the classification-based algorithm of Cardoso & da Costa [11] and the large-margin algorithm of Shashua & Levin [6] – have been accompanied by some form of generalization analysis; unfortunately, in both cases, the analysis is either incorrect or incomplete. Cardoso & da Costa (in an appendix of [11]) attempt to derive a margin-based bound for their algorithm by applying a bound for binary classifiers; however it is not clear to what function class the bound is applied, and on closer inspection it becomes clear that the analysis is, in fact, incorrect. Shashua & Levin (in [15]) also attempt to derive a margin-based bound for their algorithm; again, the quantities involved are not clearly defined, and furthermore the analysis claims to bound the 'probability that a test example will not be separated correctly' which, as we argue in Section 2, is not the natural quantity of interest in ordinal regression, and so this analysis too appears, at best, to be incomplete. These failed attempts – as well as the lack of any theoretical analysis for the other algorithms – all point to the need for a more careful analysis of the generalization properties of ordinal regression algorithms. This is what we aim to achieve in this paper.

1.2 Our Results

We formalize the mathematical setting involved in studying generalization properties of ordinal regression algorithms, including clear definitions of the quantities involved (Section 2), and then proceed to derive generalization bounds for some of these algorithms. We start with the most basic algorithms that work by learning a real-valued function in a regression framework and then rounding off a predicted real value to the closest discrete label; we relate the ordinal regression error of the resulting prediction rule to the regression error of the learned real-valued function, and use this to derive some basic 'black-box' generalization bounds for such algorithms (Section 3). Next we employ a direct stability analysis for such algorithms (Section 4); this gives better bounds in some cases. We also investigate the use of stability analysis for more general algorithms for ordinal regression, and outline some difficulties involved in achieving this goal (Section 5). Finally, we derive a margin-based bound for the state-of-the-art ordinal regression algorithm of Chu & Keerthi [10] (Section 6).

2 The Ordinal Regression Problem

The setting of the ordinal regression problem can be described as follows. There is an instance space X from which instances are drawn, and a finite set of discrete labels that have a total order relation among them; we take this set to be $[r] = \{1, \ldots, r\}$ for some $r \in \mathbb{N}$, with the usual 'greater than' order relation among the labels. The learner is given a sequence of labeled training examples $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times [r])^m$, and the goal is to learn a rule $g : X \to [r]$ that predicts accurately labels of future instances. For example, consider a user-preference modeling task in which a user gives ratings to the books she reads – ranging from 1 to 5 stars – and the goal is to predict her ratings on new books. In this case the ratings can be viewed as a discrete set of labels, but these labels clearly have an ordering among them, and so this is an instance of an ordinal regression problem (with $r = 5$).

Ordinal regression shares properties of both classification and regression: as in (multiclass) classification, the goal is to assign one of r different labels to a new instance; but as in regression, the labels are ordered, suggesting that predictions farther from the true label should incur a greater penalty than predictions closer to the true label. Indeed, if a book is rated by a user as having 5 stars, then a prediction of 4 stars would clearly be preferable to a prediction of 1 star (and should therefore incur a smaller penalty). As in classification and regression, it is generally assumed that all examples (x, y) (both training examples and future, unseen examples) are drawn randomly and independently according to some (unknown) distribution $\mathcal{D}$ on $X \times [r]$.

There are many ways to evaluate the quality of a prediction rule $g : X \to [r]$; indeed, some recent work has focused on comparing different evaluation criteria for ordinal regression [12]. In applications where the relative ranking of instances is important, it may be appropriate to consider the performance of g on pairs of examples $(x, y), (x', y')$, and count a mistake if the relative order of the predicted labels $g(x), g(x')$ differs from the relative order of the true labels y, y', *i.e.*, if $(y - y')(g(x) - g(x')) < 0$. This leads to the following 'pair-wise' error for evaluating g:

$$L_{\mathcal{D}}^{\text{pairs}}(g) = \mathbf{E}_{((x,y),(x',y'))\sim\mathcal{D}\times\mathcal{D}}\left[\mathbf{I}_{\{(y-y')(g(x)-g(x'))<0\}}\right], \tag{1}$$

where $\mathbf{I}_\phi$ denotes the indicator variable whose value is 1 if ϕ is true and 0 otherwise; this is simply the probability that g incurs a mistake on a random pair of examples, each drawn independently according to $\mathcal{D}$. As discussed in Section 1, this is the evaluation criterion used by Herbrich et al. [2]. This criterion focuses on the relative order of instances in the ranking induced by g, and is similar to the criterion used in a form of ranking problem termed r-partite ranking (see, for example, [16]).

In the setting we consider, however, the goal is not to produce a ranking of instances, but rather to predict a label for each instance that is as close as possible to the true label; in other words, we are interested in the performance of g on *individual* examples. Again, there are several ways of measuring the loss of a prediction rule g on an example (x, y) depending on the particular application and the semantics associated with the labels. A common approach, which has been used explicitly or implicitly by a majority of the more recent papers on ordinal regression, is to use the absolute loss $\ell_{\text{ord}}(g, (x, y)) = |g(x) - y|$ – henceforth referred to as the *ordinal regression loss* –

which simply measures the absolute difference between the predicted and true labels.[2] This is the loss we use.

Thus, in the setting considered in this paper, the quality of a prediction rule $g : X \to [r]$ is measured by its *expected ordinal regression error* with respect to $\mathcal{D}$:

$$L_{\mathcal{D}}^{\text{ord}}(g) = \mathbf{E}_{(x,y)\sim\mathcal{D}}\left[|g(x) - y|\right] . \tag{2}$$

In practice, since the distribution $\mathcal{D}$ is not known, the expected ordinal regression error of a prediction rule g cannot be computed exactly; instead, it must be estimated using an empirically observable quantity, such as the *empirical ordinal regression error* of g with respect to a finite sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times [r])^m$:

$$\widehat{L}_S^{\text{ord}}(g) = \frac{1}{m}\sum_{i=1}^{m} |g(x_i) - y_i| . \tag{3}$$

Our goal in this paper is to derive generalization bounds for ordinal regression algorithms; in particular, we wish to derive probabilistic bounds on the expected ordinal regression error of a learned prediction rule in terms of an empirical quantity, such as the empirical error of the rule, measured with respect to the training sample from which it is learned.

3 Black-Box Bounds for Regression-Based Algorithms

We start with the most basic algorithms that learn a real-valued function $f : X \to \mathbb{R}$ in a standard regression setting, treating the labels y_i simply as real values, and then round off a predicted real value $f(x)$ to the closest label in $[r]$; the prediction rule for such an algorithm is given by

$$g_f(x) = \begin{cases} 1, & \text{if } \ f(x) < 1 + \frac{1}{2} \\ j, & \text{if } \ j - \frac{1}{2} \leq f(x) < j + \frac{1}{2}, \quad j \in \{2, \ldots, r-1\} \\ r, & \text{if } \ f(x) \geq r - \frac{1}{2} , \end{cases} \tag{4}$$

which can also be written as

$$g_f(x) = \min_{j\in\{1,\ldots,r\}} \{j : f(x) < b^j\} , \tag{5}$$

with $b^j = j + \frac{1}{2}$ for $j \in \{1, \ldots, r-1\}$ and $b^r = \infty$. In this section, we relate the ordinal regression error of such a prediction rule g_f to the regression error of the underlying real-valued function f; this allows us to use established generalization bounds for regression algorithms to obtain some basic black-box bounds for the resulting ordinal regression algorithms.

[2] While many of the ordinal regression papers discussed in Section 1 (including [14, 6, 5] which, despite the the term 'ranking' in their titles, are on ordinal regression) explicitly employ the absolute loss, several others (such as [7, 11]) use this loss implicitly – in the form of the mean absolute error (MAE) or mean absolute distance (MAD) criterion – when measuring performance empirically on benchmark data sets.

The loss of a real-valued function $f : X \to \mathbb{R}$ on an example $(x, y) \in X \times \mathbb{R}$ in the regression setting is usually measured either by the absolute loss $\ell_{\text{abs}}(f, (x, y)) = |f(x) - y|$, or more frequently, by the squared loss $\ell_{\text{sq}}(f, (x, y)) = (f(x) - y)^2$. The quality of f with respect to a distribution $\mathcal{D}$ on $X \times \mathbb{R}$ is then measured by either its expected absolute error or its expected squared error:

$$L_{\mathcal{D}}^{\text{abs}}(f) = \mathbf{E}_{(x,y)\sim\mathcal{D}}\left[|f(x) - y|\right]; \qquad L_{\mathcal{D}}^{\text{sq}}(f) = \mathbf{E}_{(x,y)\sim\mathcal{D}}\left[(f(x) - y)^2\right]. \quad (6)$$

The corresponding empirical quantities with respect to $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times \mathbb{R})^m$ are defined analogously:

$$\widehat{L}_S^{\text{abs}}(f) = \frac{1}{m}\sum_{i=1}^{m} |f(x_i) - y_i|; \qquad \widehat{L}_S^{\text{sq}}(f) = \frac{1}{m}\sum_{i=1}^{m} (f(x_i) - y_i)^2. \quad (7)$$

The following simple lemma provides a connection between the respective errors of a real-valued function f and the corresponding prediction rule g_f.

Lemma 1. *For all $f : X \to \mathbb{R}$ and for all $(x, y) \in X \times [r]$, we have:*

1. $|g_f(x) - y| \leq \min\left(|f(x) - y| + \frac{1}{2},\ 2|f(x) - y|\right)$.
2. $|g_f(x) - y| \leq \min\left(2(f(x) - y)^2 + \frac{1}{2},\ 4(f(x) - y)^2\right)$.

Proof. If $|g_f(x) - y| = 0$, both results clearly hold. Therefore assume $|g_f(x) - y| \neq 0$. Then $|g_f(x) - y| \in \{1, \ldots, r - 1\}$, and from the definition of g_f, it follows that

$$|f(x) - y| \geq \frac{1}{2}. \quad (8)$$

Part 1. The definition of g_f easily yields the first inequality:

$$|g_f(x) - y| \leq |f(x) - y| + \frac{1}{2}. \quad (9)$$

Combining this with Eq. (8) gives the second inequality:

$$|g_f(x) - y| \leq 2|f(x) - y|. \quad (10)$$

Part 2. From Eq. (8), we have $2|f(x) - 1| \geq 1$. Since $a \geq 1 \Rightarrow a \leq a^2$, this gives

$$2|f(x) - 1| \leq 4(f(x) - 1)^2.$$

Combining this with Eqs. (9) and (10) yields the desired inequalities. □

Theorem 1. *For all $f : X \to \mathbb{R}$ and for all distributions $\mathcal{D}$ on $X \times [r]$, we have:*

1. $L_{\mathcal{D}}^{\text{ord}}(g_f) \leq \phi(L_{\mathcal{D}}^{\text{abs}}(f))$, *where* $\phi(L) = \min\left(L + \frac{1}{2},\ 2L\right)$.
2. $L_{\mathcal{D}}^{\text{ord}}(g_f) \leq \psi(L_{\mathcal{D}}^{\text{sq}}(f))$, *where* $\psi(L) = \min\left(2L + \frac{1}{2},\ 4L\right)$.

Proof. Immediate from Lemma 1. □

As a consequence of Theorem 1, any generalization result that provides a bound on the expected (absolute or squared) error of the real-valued function f learned by a regression algorithm immediately provides a bound also on the expected ordinal regression error of the corresponding prediction rule g_f. Below we provide two specific examples of such black-box bounds: the first uses a standard uniform convergence bound for regression algorithms that is expressed in terms of covering numbers; the second uses a stability bound for regression algorithms due to Bousquet & Elisseeff [17].

Theorem 2 (Covering number bound). *Let $\mathcal{F}$ be a class of real-valued functions on X, and let $\mathcal{A}$ be an ordinal regression algorithm which, given as input a training sample $S \in (X \times [r])^m$, learns a real-valued function $f_S \in \mathcal{F}$ and returns as output the prediction rule $g_S \equiv g_{f_S}$. Then for any $\varepsilon > 0$ and for any distribution $\mathcal{D}$ on $X \times [r]$,*

$$\mathbf{P}_{S\sim\mathcal{D}^m}\left[L^{\text{ord}}_{\mathcal{D}}(g_S) \;\leq\; \psi\left(\widehat{L}^{\text{sq}}_S(f_S)+\varepsilon\right)\right] \;\geq\; 1 - 4\mathcal{N}_1(\varepsilon/16,\mathcal{F},2m)\cdot\exp(-m\varepsilon^2/32)\,,$$

where $\psi(\cdot)$ is as defined in Theorem 1, and $\mathcal{N}_1$ refers to d_1 covering numbers.

Proof. The following bound on the expected squared error of the learned real-valued function is well known (cf. the uniform convergence result in Theorem 17.1 of [18]):

$$\mathbf{P}_{S\sim\mathcal{D}^m}\left[L^{\text{sq}}_{\mathcal{D}}(f_S) \;\leq\; \widehat{L}^{\text{sq}}_S(f_S)+\varepsilon\right] \;\geq\; 1 - 4\mathcal{N}_1(\varepsilon/16,\mathcal{F},2m)\cdot\exp(-m\varepsilon^2/32)\,.$$

The result then follows from Part 2 of Theorem 1. □

Next we review some notions needed to present the stability bound.

Definition 1 (Loss stability). *Let $\ell(f,(x,y))$ be a loss function defined for $f : X\to\mathbb{R}$ and $(x,y) \in X \times Y$ for some $Y \subseteq \mathbb{R}$. A regression algorithm whose output on a training sample $S = ((x_1,y_1),\ldots,(x_m,y_m)) \in (X\times Y)^m$ we denote by $f_S : X\to\mathbb{R}$ is said to have* loss stability *β with respect to ℓ (where $\beta : \mathbb{N}\to\mathbb{R}$) if for all $m \in \mathbb{N}$, $S \in (X\times Y)^m$, $1 \leq i \leq m$ and $(x_i',y_i') \in X \times Y$, we have for all $(x,y) \in X \times Y$,*

$$|\ell(f_S,(x,y)) - \ell(f_{S^i},(x,y))| \leq \beta(m)\,,$$

where S^i denotes the sequence obtained from S by replacing (x_i,y_i) with (x_i',y_i').

Theorem 3 ([17][3]). *Let $\ell(f,(x,y))$ be a loss function defined for $f : X\to\mathbb{R}$ and $(x,y) \in X \times Y$ for some $Y \subseteq \mathbb{R}$. Let $\mathcal{A}$ be a regression algorithm which, given as input a training sample $S \in (X\times Y)^m$, returns as output a real-valued function $f_S : X\to\mathbb{R}$, such that $0 \leq \ell(f_S,(x,y)) \leq M$ for all S and all $(x,y) \in X \times Y$. If $\mathcal{A}$ has loss stability β with respect to ℓ, then for any $0 < \delta < 1$ and for any distribution $\mathcal{D}$ on $X \times Y$, with probability at least $1-\delta$ over the draw of S (according to $\mathcal{D}^m$),*

$$L^{\ell}_{\mathcal{D}}(f_S) \;\leq\; \widehat{L}^{\ell}_S(f_S) + \beta(m) + \big(2m\beta(m)+M\big)\sqrt{\frac{\ln(1/\delta)}{2m}}\,,$$

where $L^{\ell}_{\mathcal{D}}$, $\widehat{L}^{\ell}_S$ denote expected and empirical ℓ-errors, defined analogously to (6–7).

[3] The version presented here differs slightly (only in constants) from the result of [17]. This is due to a slight difference in definitions of stability: our definitions are in terms of changes to a training sample that consist of replacing one element in the sample with a new one, while those in [17] are in terms of changes that consist of removing one element from the sample.

Theorem 4 (Stability bound). *Let $\mathcal{A}$ be an ordinal regression algorithm which, given as input a training sample $S \in (X \times [r])^m$, learns a real-valued function $f_S : X \to [c, d]$ using a regression algorithm that has loss stability β with respect to the squared loss ℓ_{sq} (defined for $(x, y) \in X \times [r]$), and returns as output the prediction rule $g_S \equiv g_{f_S}$. Then for any $0 < \delta < 1$ and for any distribution $\mathcal{D}$ on $X \times [r]$, with probability at least $1 - \delta$ over the draw of S (according to $\mathcal{D}^m$),*

$$L_{\mathcal{D}}^{\text{ord}}(g_S) \;\leq\; \psi\left(\widehat{L}_S^{\text{sq}}(f_S) + \beta(m) + \big(2m\beta(m) + M\big)\sqrt{\frac{\ln(1/\delta)}{2m}}\right),$$

where $M = (\max(d, r) - \min(c, 1))^2$, and $\psi(\cdot)$ is as defined in Theorem 1.

Proof. Follows from Theorem 3 applied to ℓ_{sq} (note that $0 \leq \ell_{\text{sq}}(f, (x, y)) \leq M$ for all $f : X \to [c, d]$ and all $(x, y) \in X \times [r]$), and Part 2 of Theorem 1. □

Example 1 (Bound 1 for SVR-based algorithm). As a further example of how Theorem 1 can be applied, consider an ordinal regression algorithm which, given a training sample $S \in (X \times [r])^m$, uses the support vector regression (SVR) algorithm to learn a real-valued function $f_S \in \mathcal{F}$ in some reproducing kernel Hilbert space (RKHS) $\mathcal{F}$, and returns the prediction rule $g_S \equiv g_{f_S}$. The SVR algorithm minimizes a regularized version of the empirical ℓ_ϵ-error $\widehat{L}_S^\epsilon(f) = \frac{1}{m}\sum_{i=1}^m \ell_\epsilon(f, (x_i, y_i))$ for some $\epsilon > 0$, where ℓ_ϵ is the ϵ-insensitive loss defined by $\ell_\epsilon(f, (x, y)) = (|f(x) - y| - \epsilon)_+$ (here $a_+ = \max(a, 0)$). If the kernel K associated with $\mathcal{F}$ satisfies $K(x, x) \leq \kappa^2 \; \forall x \in X$, and a regularization parameter λ is used, then the SVR algorithm has loss stability $2\kappa^2/\lambda m$ with respect to ℓ_ϵ [17], and furthermore, with $M = \max\left(r + \kappa\sqrt{r/\lambda},\; 2\kappa\sqrt{r/\lambda}\right)$, satisfies $0 \leq \ell_\epsilon(f_S, (x, y)) \leq M$. Therefore, applying Theorem 3 to ℓ_ϵ, observing that $\ell_{\text{abs}} \leq \ell_\epsilon + \epsilon$, and finally, using Part 1 of Theorem 1, we get that for any $0 < \delta < 1$ and for any distribution $\mathcal{D}$ on $X \times [r]$, with probability at least $1 - \delta$ over $S \sim \mathcal{D}^m$,

$$L_{\mathcal{D}}^{\text{ord}}(g_S) \;\leq\; \phi\left(\widehat{L}_S^\epsilon(f_S) + \epsilon + \frac{2\kappa^2}{\lambda m} + \left(\frac{4\kappa^2}{\lambda} + M\right)\sqrt{\frac{\ln(1/\delta)}{2m}}\right), \tag{11}$$

where $\phi(\cdot)$ is as defined in Theorem 1.

4 Direct Stability Analysis for Regression-Based Algorithms

The stability bounds for regression-based algorithms discussed above – in Theorem 4 and in Example 1 – make use of existing stability bounds for regression algorithms in a black-box manner. In this section, we directly analyze the stability of these algorithms in the context of the ordinal regression error of the final prediction rule; this allows us to obtain alternative bounds which in some cases are tighter than those obtained through the above black-box analysis. We start with an alternative definition of the stability of a regression algorithm (note that the algorithms we consider here are the same as before, *i.e.*, algorithms that learn a real-valued function f in a regression setting and then make label predictions according to g_f; the difference will lie in the manner in which we analyze these algorithms). The approach we use is similar to that used by Bousquet & Elisseeff [17] to analyze binary classification algorithms that learn a real-valued function f and then make class predictions according to $\text{sgn}(f)$.

Definition 2 (Score stability). *A regression algorithm whose output on a training sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times \mathbb{R})^m$ we denote by $f_S : X \to \mathbb{R}$ is said to have* score stability *ν (where $\nu : \mathbb{N} \to \mathbb{R}$) if for all $m \in \mathbb{N}$, $S \in (X \times Y)^m$, $1 \leq i \leq m$ and $(x_i', y_i') \in X \times Y$, we have for all $x \in X$,*

$$|f_S(x) - f_{S^i}(x)| \leq \nu(m)\,,$$

where S^i denotes the sequence obtained from S by replacing (x_i, y_i) with (x_i', y_i').

The following loss, defined for $f : X \to \mathbb{R}$ and, crucially, for $(x, y) \in X \times [r]$, will play an important role in our analysis; we refer to it as the 'clipped' loss:

$$\ell_{\text{clip}}(f, (x, 1)) = \begin{cases} 0, & \text{if } f(x) < 1 \\ 2(f(x) - 1), & \text{if } 1 \leq f(x) < \frac{r+1}{2} \\ r - 1, & \text{if } f(x) \geq \frac{r+1}{2}; \end{cases}$$

$$\ell_{\text{clip}}(f, (x, y)) = \begin{cases} y - 1, & \text{if } f(x) < \frac{y+1}{2} \\ 2(y - f(x)), & \text{if } \frac{y+1}{2} \leq f(x) < y \\ 2(f(x) - y), & \text{if } y \leq f(x) < r - \frac{y+r}{2} \\ r - y, & \text{if } f(x) \geq \frac{y+r}{2} \end{cases} \quad \text{for } y \in \{2, \ldots, r-1\};$$

$$\ell_{\text{clip}}(f, (x, r)) = \begin{cases} r - 1, & \text{if } f(x) < \frac{r+1}{2} \\ 2(r - f(x)), & \text{if } \frac{r+1}{2} \leq f(x) < r \\ 0, & \text{if } f(x) \geq r. \end{cases}$$

Figure 1 shows plots of this loss for $r = 4$. A crucial property of this loss, which is immediate from the definitions (see Figure 1), is the following:

Lemma 2. *For all $f : X \to \mathbb{R}$ and for all $(x, y) \in X \times [r]$, we have:*

$$|g_f(x) - y| \;\leq\; \ell_{\text{clip}}(f, (x, y)) \;\leq\; 2|f(x) - y|\,.$$

The following lemma shows that a regression algorithm that has good score stability also has good loss stability with respect to ℓ_{clip}. The proof is similar to the proof of Lemma 2 of [19], and is omitted for lack of space.

Lemma 3. *If a real-valued function learning algorithm has score stability ν (where $\nu : \mathbb{N} \to \mathbb{R}$), then it has loss stability $\beta = 2\nu$ with respect to the clipped loss ℓ_{clip}.*

We are now ready for the main result of this section:

Theorem 5 (Direct stability bound). *Let $\mathcal{A}$ be an ordinal regression algorithm which, given as input a training sample $S \in (X \times [r])^m$, learns a real-valued function $f_S : X \to \mathbb{R}$ using a regression algorithm that has score stability ν, and returns as output the prediction rule $g_S \equiv g_{f_S}$. Then for any $0 < \delta < 1$ and for any distribution $\mathcal{D}$ on $X \times [r]$, with probability at least $1 - \delta$ over the draw of S (according to $\mathcal{D}^m$),*

$$L_{\mathcal{D}}^{\text{ord}}(g_S) \;\;\leq\;\; \widehat{L}_S^{\text{clip}}(f_S) + 2\nu(m) + \big(4m\nu(m) + r - 1\big)\sqrt{\frac{\ln(1/\delta)}{2m}}\,,$$

where $\widehat{L}_S^{\text{clip}}$ denotes the empirical ℓ_{clip}-error with respect to S.

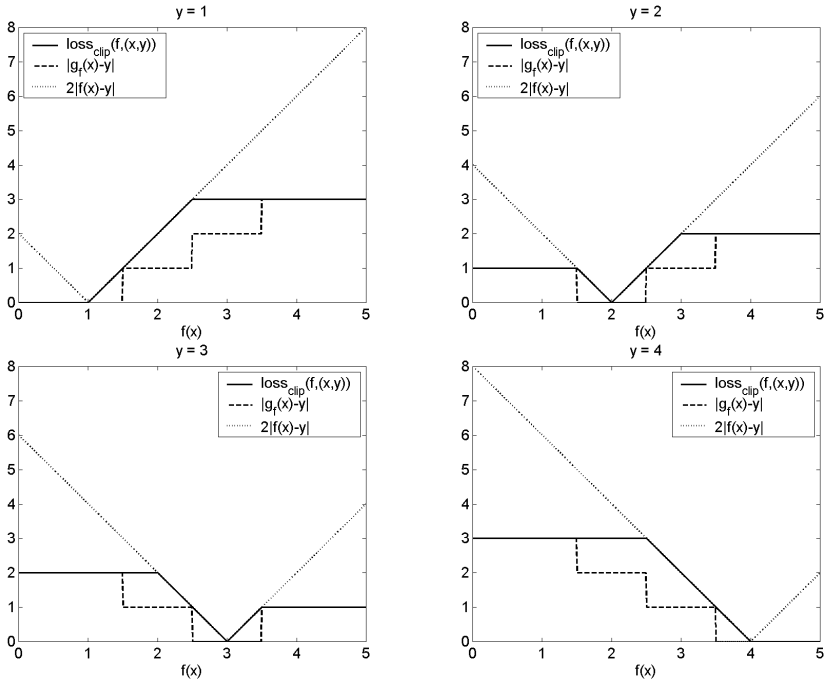

Fig. 1. Plots of the clipped loss $\ell_{\text{clip}}(f,(x,y))$, together with $\ell_{\text{ord}}(g_f,(x,y)) = |g_f(x) - y|$, and $2\ell_{\text{abs}}(f,(x,y)) = 2|f(x) - y|$, for $y = 1, 2, 3$ and 4 (each plotted as a function of $f(x)$), for an ordinal regression problem with $r = 4$.

Proof. By Lemma 3, we have that $\mathcal{A}$ has loss stability 2ν with respect to ℓ_{clip}. Furthermore, we have $0 \leq \ell_{\text{clip}}(f,(x,y)) \leq r-1$ for all $f : X \to \mathbb{R}$ and all $(x,y) \in X \times [r]$. The result then follows from Theorem 3 applied to ℓ_{clip} (with $\beta = 2\nu$ and $M = r-1$), and from Lemma 2. □

Example 2 (Bound 2 for SVR-based algorithm). For a comparison of the above bound with the black-box stability bounds of Section 3, consider again the SVR-based ordinal regression algorithm of Example 1 which, given a training sample $S \in (X \times [r])^m$, uses the SVR algorithm to learn a real-valued function $f_S \in \mathcal{F}$ in an RKHS $\mathcal{F}$, and returns the prediction rule $g_S \equiv g_{f_S}$. Under the conditions of Example 1, the SVR algorithm is known to have score stability $2\kappa^2/\lambda m$ [17]. Therefore, by Theorem 5, we get that for any $0 < \delta < 1$ and for any distribution $\mathcal{D}$ on $X \times [r]$, with probability at least $1-\delta$ over $S \sim \mathcal{D}^m$,

$$L_{\mathcal{D}}^{\text{ord}}(g_S) \;\leq\; \widehat{L}_S^{\text{clip}}(f_S) + \frac{4\kappa^2}{\lambda m} + \left(\frac{8\kappa^2}{\lambda} + r - 1\right)\sqrt{\frac{\ln(1/\delta)}{2m}}\,. \tag{12}$$

Comparing this with the bound of Example 1, we find that if the term in the argument of $\phi(\cdot)$ in Eq. (11) is smaller than $1/2$, then the direct bound above is guaranteed to give a tighter generalization result.

5 Stability Analysis for More General Algorithms

So far we have considered regression-based algorithms for ordinal regression that learn a real-valued function $f : X \to \mathbb{R}$ and then make label predictions according to g_f. We now consider more general algorithms that learn both a real-valued function $f : X \to \mathbb{R}$ and a set of thresholds $b^1 \leq \ldots \leq b^{r-1} \leq b^r = \infty$; as discussed previously, the prediction rule in this case is given by

$$g_{f,b}(x) = \min_{j \in \{1,\ldots,r\}} \{j : f(x) < b^j\}\,, \tag{13}$$

where $b \equiv (b^1, \ldots, b^{r-1})$ denotes the vector of $r-1$ thresholds (note that b^r is fixed to ∞). Recall that regression-based algorithms can be viewed as using a fixed threshold vector given by $b^j = j + \frac{1}{2}$ for $j \in \{1, \ldots, r-1\}$. In what follows, the term threshold vector will always refer to a vector of thresholds $(b^1, \ldots, b^{r-1})$ that satisfies the inequalities $b_1 \leq \ldots \leq b_{r-1}$.

The ordinal regression loss of a prediction rule $g_{f,b}$ on an example $(x, y) \in X \times [r]$ effectively counts the number of thresholds b^j such that $f(x)$ falls to the wrong side of b^j:

$$\left|g_{f,b}(x) - y\right| \;=\; \sum_{j=1}^{y-1} \mathbf{I}_{\{f(x) < b^j\}} + \sum_{j=y}^{r-1} \mathbf{I}_{\{f(x) \geq b^j\}}\,. \tag{14}$$

In general, the loss on an example $(x, y) \in X \times [r]$ in this more general setting is determined by both f and b, and as before, given a loss function $\ell(f, b, (x, y))$ in this setting, we can define the expected and empirical ℓ-errors of a function/threshold-vector pair (f, b) with respect to a distribution $\mathcal{D}$ on $X \times [r]$ and a sample $S \in (X \times [r])^m$, respectively, as follows:

$$L^\ell_{\mathcal{D}}(f, b) \;=\; \mathbf{E}_{(x,y)\sim\mathcal{D}}\big[\ell(f, b, (x, y))\big]\,; \quad \widehat{L}^\ell_S(f, b) \;=\; \frac{1}{m}\sum_{i=1}^{m} \ell(f, b, (x_i, y_i))\,. \tag{15}$$

In order to analyze ordinal regression algorithms in this more general setting, we can extend the notion of loss stability in a straightforward manner to loss functions $\ell(f, b, (x, y))$. It is then possible to show the following result, which states that an algorithm with good loss stability with respect to such a loss ℓ has good generalization properties with respect to the error induced by ℓ. We omit the proof, which follows the proofs of similar results for classification/regression and ranking in [17, 19].

Theorem 6 (Stability bound for (f, b)-learners). *Let $\mathcal{A}$ be an ordinal regression algorithm which, given as input a training sample $S \in (X \times [r])^m$, learns a real-valued function $f_S : X \to \mathbb{R}$ and a threshold vector $b_S \equiv (b^1_S, \ldots, b^{r-1}_S)$, and returns as output the prediction rule $g_S \equiv g_{f_S, b_S}$. Let ℓ be any loss function in this setting such that $0 \leq \ell(f_S, b_S, (x, y)) \leq M$ for all training samples S and all $(x, y) \in X \times [r]$, and let $\beta : \mathbb{N} \to \mathbb{R}$ be such that $\mathcal{A}$ has loss stability β with respect to ℓ. Then for any $0 < \delta < 1$ and for any distribution $\mathcal{D}$ on $X \times [r]$, with probability at least $1 - \delta$ over the draw of S (according to $\mathcal{D}^m$),*

$$L^\ell_{\mathcal{D}}(f_S, b_S) \;\leq\; \widehat{L}^\ell_S(f_S, b_S) + \beta(m) + \big(2m\beta(m) + M\big)\sqrt{\frac{\ln(1/\delta)}{2m}}\,.$$

In the case of classification and regression, and also of ranking, once a stability-based generalization result of the above form was established, it was quickly shown that there were practical learning algorithms for those problems that satisfied the desired stability conditions, and hence the result could be applied to these algorithms to obtain generalization bounds for them (indeed, we used such bounds for the support vector regression (SVR) algorithm in our study of regression-based algorithms in the previous two sections). Unfortunately, this has proved to be more difficult in our setting.

Given that many of the classification, regression and ranking algorithms for which stability analysis has been successful are regularization-based algorithms – with particular success among algorithms that perform regularization in an RKHS (such as the SVR algorithm for regression or SVMs for classification) – a natural candidate for stability analysis in our setting is the ordinal regression algorithm of Chu & Keerthi [10], which is inspired by SVMs and also performs regularization in an RKHS. However, our attempts to show stability of this algorithm have so far had limited success.

The algorithm of [10] learns a real-valued function f and a threshold vector b by minimizing a regularized upper bound on the empirical ordinal regression error. Specifically, if we associate with each label $y \in [r]$ the sign vector $(y^1, \ldots, y^{r-1}) \in \{-1, +1\}^{r-1}$ defined by

$$y^j \;=\; \begin{cases} +1, \text{ if } j \in \{1, \ldots, y-1\} \\ -1, \text{ if } j \in \{y, \ldots, r-1\}, \end{cases} \tag{16}$$

then the loss ℓ_{CK} used by Chu & Keerthi is given by

$$\ell_{\text{CK}}(f, b, (x, y)) \;=\; \sum_{j=1}^{r-1} \left(1 - y^j (f(x) - b^j)\right)_+ . \tag{17}$$

Comparing with Eq. (14), it is easily verified that

$$\left| g_{f,b}(x) - y \right| \;\leq\; \ell_{\text{CK}}(f, b, (x, y)) \,. \tag{18}$$

Given a training sample $S \in (X \times [r])^m$, the Chu-Keerthi algorithm returns a real-valued function $f_S \in \mathcal{F}$ and a threshold vector $b_S \in \mathbb{R}^{r-1}$ that satisfy[4]

$$(f_S, b_S) = \arg \min_{(f,b) \in \mathcal{F} \times \mathbb{R}^{r-1}} \widehat{L}_S^{\text{CK}}(f, b) + \lambda \left(\|f\|_K^2 + \|b\|^2 \right), \tag{19}$$

where $\mathcal{F}$ is an RKHS with kernel K, $\|f\|_K$ denotes the RKHS norm of f, and $\lambda > 0$ is a regularization parameter; as discussed in [10], the vector b_S returned by the above algorithm always satisfies the necessary inequalities $b_S^1 \leq \ldots \leq b_S^{r-1}$. The difficulty in showing stability of the above algorithm seems to stem from the lack of a satisfactory notion of the loss ℓ_{CK} being 'jointly convex' in $f(x)$ and the b_j; in the corresponding analysis for classification/regression and ranking algorithms, convexity of the relevant loss functions in $f(x)$ allowed the desired stability conditions to be established. This difficulty appears to apply also in considering other notions of stability, such as possible extensions of score stability to the above setting.

[4] The version here includes the regularization term for b suggested in a footnote of [10].

6 Margin Bound for Chu & Keerthi's Algorithm

We consider now a different approach to analyzing ordinal regression algorithms that learn both a real-valued function $f : X \to \mathbb{R}$ and a set of thresholds $b^1 \leq \ldots \leq b^{r-1} \leq b^r = \infty$, and then make label predictions according to the prediction rule $g_{f,b}$ defined in Eq. (13) (recall that b is the threshold vector $(b^1, \ldots, b^{r-1})$, and that the term threshold vector always refers to a vector of thresholds satisfying the above inequalities). In particular, we define a notion of margin for prediction rules of this form, and use this notion to derive generalization bounds for such algorithms. Our approach generalizes the margin-based analysis used in the study of classification algorithms, and results in a margin-based bound for the ordinal regression algorithm of Chu & Keerthi [10] discussed in Section 5.

Definition 3 (Margin). *Let $f : X \to \mathbb{R}$ and let $b \equiv (b^1, \ldots, b^{r-1})$ be a threshold vector. Then for each $j \in \{1, \ldots, r-1\}$, we define the* margin *of f with respect to b^j on an example $(x, y) \in X \times [r]$ as follows:*

$$\rho^j(f, b^j, (x, y)) \;=\; y^j(f(x) - b^j)\,,$$

where y^j is as defined in Eq. (16).

Next, for $\gamma > 0$, we define the *γ-margin loss* of a real-valued function f and a threshold vector b on an example $(x, y) \in X \times [r]$ as follows:

$$\ell_\gamma(f, b, (x, y)) \;=\; \sum_{j=1}^{r-1} \mathbf{I}_{\{\rho^j(f, b^j, (x,y)) \leq \gamma\}}\,. \tag{20}$$

The ℓ_γ loss counts the number of thresholds b^j for which the corresponding margin $\rho^j(f, b^j, (x, y))$ is smaller than (or equal to) γ; thus, comparing with Eq. (14), we immediately have that for all $\gamma > 0$,

$$\left|g_{f,b}(x) - y\right| \;\leq\; \ell_\gamma(f, b, (x, y))\,. \tag{21}$$

We then have the following margin-based generalization bound for (f, b)-learners. The proof makes use of a margin-based bound for binary classifiers (cf. Theorem 10.1 of [18]), applied separately to each of the $r - 1$ classification tasks of predicting y^j through $\mathrm{sgn}(f(x) - b^j)$; a union bound argument then leads to the result below. We omit the details due to lack of space.

Theorem 7 (Margin bound). *Let $\mathcal{F}$ be a class of real-valued functions on X, and let $\mathcal{A}$ be an ordinal regression algorithm which, given as input a training sample $S \in (X \times [r])^m$, learns a real-valued function $f_S \in \mathcal{F}$ and a threshold vector $b_S \in [-B, B]^{r-1}$, and returns as output the prediction rule $g_S \equiv g_{f_S, b_S}$. Let $\gamma > 0$. Then for any $0 < \delta < 1$ and for any distribution $\mathcal{D}$ on $X \times [r]$, with probability at least $1 - \delta$ over the draw of S (according to $\mathcal{D}^m$),*

$$L^{\mathrm{ord}}_{\mathcal{D}}(g_S) \;\leq\; \widehat{L}^\gamma_S(f_S, b_S) + (r-1)\sqrt{\frac{8}{m}\left(\ln \mathcal{N}_\infty(\gamma/2, \mathcal{F}, 2m) + \ln\left(\frac{4B(r-1)}{\delta\gamma}\right)\right)}\,,$$

where $\widehat{L}^\gamma_S$ denotes the empirical ℓ_γ-error, and $\mathcal{N}_\infty$ refers to d_∞ covering numbers.

Example 3 (Bound for Chu-Keerthi algorithm). Recall that the ordinal regression algorithm of Chu & Keerthi [10], described in Section 5, performs regularization in an RKHS $\mathcal{F}$ with kernel K as follows: given a training sample $S \in (X \times [r])^m$, the algorithm selects a function $f_S \in \mathcal{F}$ and a threshold vector b_S that minimize a regularized upper bound on the ordinal regression error of the resulting prediction rule $g_S \equiv g_{f_S,b_S}$. It is easy to show that the output of the Chu-Keerthi algorithm always satisfies

$$\|f_S\|_K^2 + \|b_S\|^2 \leq \frac{r-1}{\lambda}\,,$$

where λ is the regularization parameter. Thus we have that

$$b_S \in \left[-\sqrt{\frac{r-1}{\lambda}}, \sqrt{\frac{r-1}{\lambda}}\right]^{r-1}; \quad f_S \in \mathcal{F}_{r,\lambda} \equiv \left\{ f \in \mathcal{F} \;\middle|\; \|f\|_K^2 \leq \frac{r-1}{\lambda} \right\}.$$

By Theorem 7, it follows that if the covering numbers $\mathcal{N}_\infty(\gamma/2, \mathcal{F}_{r,\lambda}, 2m)$ of the effective function class $\mathcal{F}_{r,\lambda}$ can be upper bounded appropriately, then we have a generalization bound for the Chu-Keerthi algorithm. Such covering number bounds are known in a variety of settings. For example, if the kernel K satisfies $K(x,x) \leq \kappa^2 \; \forall \, x \in X$, then using a covering number bound of Zhang [20], we get that there is a constant C such that for any $\gamma > 0$, any $0 < \delta < 1$ and any distribution $\mathcal{D}$ on $X \times [r]$, with probability at least $1-\delta$ over $S \sim \mathcal{D}^m$,

$$L_{\mathcal{D}}^{\mathrm{ord}}(g_S) \;\; \leq \;\; \widehat{L}_S^{\gamma}(f_S, b_S) + (r-1)\sqrt{\frac{C}{m}\left(\frac{\kappa^2(r-1)}{\lambda\gamma^2}\ln(m) + \ln\left(\frac{r-1}{\lambda\delta\gamma}\right)\right)}\,.$$

7 Conclusion

Our goal in this paper has been to study generalization properties of ordinal regression algorithms that learn to predict labels in a discrete but ordered set. We have focused on the absolute loss $|g(x) - y|$, for which we have obtained bounds in a variety of settings; other losses such as the squared loss $(g(x) - y)^2$ can also be useful and should be explored. Note that all such losses that measure the performance of a prediction rule g on a single example (x, y) must necessarily assume a metric on the set of labels y; in our case, we assume the labels are in $\{1, \ldots, r\}$, with the absolute distance metric (such labels are referred to as having an *interval* scale in [1]). In applications where the labels are ordered but cannot be associated with a metric, it may be more appropriate to consider losses that measure the ranking performance of g on pairs of examples [2, 16].

Another important question concerns the consistency properties of ordinal regression algorithms: whether they converge to an optimal solution, and if so, at what rate. It would be particularly interesting to study the consistency properties of algorithms that minimize a convex upper bound on the ordinal regression error, as has been done recently for classification [21, 22].

Acknowledgments

We would like to thank Yoram Singer for discussions on ordinal regression and for pointing us to the need for generalization bounds for this problem. This research was supported in part by NSF award DMS-0732334.

References

1. McCullagh, P., Nelder, J.A.: Generalized Linear Models. Second edn. Chapman and Hall (1989)
2. Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. In: Advances in Large Margin Classifiers. MIT Press (2000) 115–132
3. Kramer, S., Pfahringer, B., Widmer, G., Groeve, M.D.: Prediction of ordinal classes using regression trees. Fundamenta Informaticae **47** (2001) 1001–1013
4. Frank, E., Hall, M.: A simple approach to ordinal classification. In: Proceedings of the 12th European Conference on Machine Learning. (2001) 145–156
5. Crammer, K., Singer, Y.: Online ranking by projecting. Neural Computation **17**(1) (2005) 145–175
6. Shashua, A., Levin, A.: Ranking with large margin principle: Two approaches. In: Advances in Neural Information Processing Systems 15, MIT Press (2003) 937–944
7. Harrington, E.F.: Online ranking/collaborative filtering using the perceptron algorithm. In: Proceedings of the 20th International Conference on Machine Learning. (2003) 250–257
8. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. Journal of Machine Learning Research **6** (2005) 1019–1041
9. Rennie, J.D.M., Srebro, N.: Loss functions for preference levels: Regression with discrete ordered labels. In: Proc. IJCAI Multidisciplinary Workshop on Advances in Preference Handling. (2005)
10. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural Computation **19**(3) (2007) 792–815
11. Cardoso, J.S., da Costa, J.F.P.: Learning to classify ordinal data: The data replication method. Journal of Machine Learning Research **8** (2007) 1393–1429
12. Waegeman, W., De Baets, B., Boullart, L.: ROC analysis in ordinal regression learning. Pattern Recognition Letters **29**(1) (2008) 1–9
13. Mathieson, M.J.: Ordinal models for neural networks. In: Neural Networks in Financial Engineering. World Scientific (1996) 523–536
14. Crammer, K., Singer, Y.: Pranking with ranking. In: Advances in Neural Information Processing Systems 14, MIT Press (2002) 641–647
15. Shashua, A., Levin, A.: Taxonomy of large margin principle algorithms for ordinal regression problems. Technical Report 2002-39, Leibniz Center for Research, School of Computer Science and Engg., The Hebrew University of Jerusalem (2002)
16. Rajaram, S., Agarwal, S.: Generalization bounds for k-partite ranking. In: Proceedings of the NIPS-2005 Workshop on Learning to Rank. (2005)
17. Bousquet, O., Elisseeff, A.: Stability and generalization. Journal of Machine Learning Research **2** (2002) 499–526
18. Anthony, M., Bartlett, P.L.: Learning in Neural Networks: Theoretical Foundations. Cambridge University Press (1999)
19. Agarwal, S., Niyogi, P.: Stability and generalization of bipartite ranking algorithms. In: Proceedings of the 18th Annual Conference on Learning Theory. (2005)
20. Zhang, T.: Covering number bounds of certain regularized linear function classes. Journal of Machine Learning Research **2** (2002) 527–550
21. Zhang, T.: Statistical behavior and consistency of classification methods based on convex risk minimization. The Annals of Statistics **32** (2004) 56–85
22. Bartlett, P.L., Jordan, M.I., McAuliffe, J.D.: Convexity, classification, and risk bounds. Journal of the American Statistical Association **101**(473) (2006) 138–156