7 Large Margin Rank Boundaries for Ordinal Regression

Ralf Herbrich
ralfh@cs.tu-berlin.de

Thore Graepel
graepel2@cs.tu-berlin.de

Klaus Obermayer
oby@cs.tu-berlin.de

Technical University of Berlin
Department of Computer Science
Franklinstr. 28/29,
10587 Berlin,
Germany

In contrast to the standard machine learning tasks of classification and metric regression we investigate the problem of predicting variables of ordinal scale, a setting referred to as *ordinal regression*. This problem arises frequently in the social sciences and in information retrieval where human preferences play a major role. Whilst approaches proposed in statistics rely on a probability model of a latent (unobserved) variable we present a distribution independent risk formulation of ordinal regression which allows us to derive a uniform convergence bound. Applying this bound we present a large margin algorithm that is based on a mapping from objects to scalar utility values thus classifying pairs of objects. We give experimental results for an information retrieval task which show that our algorithm outperforms more naive approaches to ordinal regression such as Support Vector Classification and Support Vector Regression in the case of more than two ranks.

7.1 Introduction

Let us shortly recall the model presented in Chapter 1. Given an iid sample (X, Y), and a set F of mappings $f : \mathcal{X} \mapsto \mathcal{Y}$, a learning procedure aims at finding f^* such that — using a predefined loss $c : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ — the risk functional (1.26) is minimized. Using the principle of Empirical Risk Minimization (ERM), one chooses the function f_{emp} which minimizes the mean of the loss $R_{\text{emp}}(f)$ (Equation 1.27) given the sample (X, Y). Introducing a quantity which characterizes the capacity of F, bounds for the deviation $|R(f_{\text{emp}}) - \inf_{f \in F} R(f)|$ can be derived (see Theorems 1.5, 1.6, 1.10, and 1.11). Two main scenarios were considered in the past: (i) If $\mathcal{Y}$ is a finite unordered set (nominal scale), the task is referred to as classification learning. Since $\mathcal{Y}$ is unordered, the $0-1$ loss, i.e., $c_{\text{class}}(\mathbf{x}, y, f(\mathbf{x})) = 1_{f(\mathbf{x}) \neq y}$, is adequate to capture the loss at each point $(\mathbf{x}, y)$. (ii) If $\mathcal{Y}$ is a metric space, e.g., the set of real numbers, the task is referred to as regression estimation. In this case the loss function can take into account the full metric structure. Different metric loss functions have been proposed which are optimal under given probability models $P(y|\mathbf{x})$ (cf. Huber [1981]). Usually, optimality is measured in terms of the mean squared error of f_{emp}.

classification and regression

Here, we consider a problem which shares properties of both cases (i) and (ii). Like in (i) $\mathcal{Y}$ is a finite set and like in (ii) there exists an ordering among the elements of $\mathcal{Y}$. In contrast to regression estimation we have to deal with the fact that $\mathcal{Y}$ is a non–metric space. A variable of the above type exhibits an *ordinal scale* and can be considered as the result of a coarsely measured continuous variable [Anderson and Philips, 1981]. The ordinal scale leads to problems in defining an appropriate loss function for our task (see also McCullagh [1980] and Anderson [1984]): On the one hand, there exists no metric in the space $\mathcal{Y}$, i.e., the distance $(y - y')$ of two elements is not defined. On the other hand, the simple $0-1$ loss does not reflect the ordering in $\mathcal{Y}$. Since no loss function $c(\mathbf{x}, y, f(\mathbf{x}))$ can be found that acts on true ranks y and predicted ranks $f(\mathbf{x})$, we suggest to exploit the ordinal nature of the elements of $\mathcal{Y}$ by considering the order on the space $\mathcal{X}$ induced by each mapping $f : \mathcal{X} \mapsto \mathcal{Y}$. Thus our loss function $c_{\text{pref}}(\mathbf{x}_1, \mathbf{x}_2, y_1, y_2, f(\mathbf{x}_1), f(\mathbf{x}_2))$ acts on pairs of true ranks (y_1, y_2) and predicted ranks $(f(\mathbf{x}_1), f(\mathbf{x}_2))$. Such an approach makes it possible to formulate a distribution independent theory of ordinal regression and to give uniform bounds for the risk functional. Roughly speaking, the proposed risk functional measures the probability of misclassification of a randomly drawn pair $(\mathbf{x}_1, \mathbf{x}_2)$ of observations, where the two classes are $\mathbf{x}_1 \succ_{\mathcal{X}} \mathbf{x}_2$ and $\mathbf{x}_2 \succ_{\mathcal{X}} \mathbf{x}_1$ (see Section 7.3). Problems of ordinal regression arise in many fields, e.g., in information retrieval [Wong et al., 1988, Herbrich et al., 1998], in econometric models [Tangian and Gruber, 1995, Herbrich et al., 1999b], and in classical statistics [McCullagh, 1980, Fahrmeir and Tutz, 1994, Anderson, 1984, de Moraes and Dunsmore, 1995, Keener and Waldman, 1985].

distribution independent theory of ordinal regression

As an application of the above–mentioned theory, we suggest to model ranks by intervals on the real line. Then the task is to find a latent utility function that maps objects to scalar values. Due to the ordering of ranks, the function is restricted to be transitive and asymmetric, because these are the defining properties of a *preference relation*. The resulting learning task is also referred to as learning of preference relations (see Herbrich et al. [1998]). One might think that learning of preference relations reduces to a standard classification problem if pairs of objects are considered. This, however, is not true in general because the properties of transitivity and asymmetry may be violated by traditional Bayesian approaches due to the problem of stochastic transitivity [Suppes et al., 1989]. Considering pairs of objects, the task of learning reduces to finding a utility function that best reflects the preferences induced by the unknown distribution $p(\mathbf{x}, y)$. Our learning procedure on pairs of objects is an application of the large margin idea known from data–dependent Structural Risk Minimization [Shawe-Taylor et al., 1998]. The resulting algorithm is similar to Support Vector Machines (see Section 1.3). Since during learning and application of SVMs only inner products of object representations $\mathbf{x}_i$ and $\mathbf{x}_j$ have to be computed, the method of potential functions can be applied (see Aizerman et al. [1964] or Section 1.3.2).

preference relation

large margin

In Section 7.2 we introduce the setting of ordinal regression and shortly present well known results and models from the field of statistics. In Section 7.3 we introduce our model for ordinal regression and give a bound for the proposed loss function. In the following section we present an algorithm for ordinal regression based on large margin techniques. In Section 7.5 we give learning curves of our approach in a controlled experiment and in a real–world experiment on data from information retrieval.

7.2 Classical Models for Ordinal Regression

In this section we shortly recall the well–known cumulative or threshold model for ordinal regression [McCullagh and Nelder, 1983].

In contrast to Equation (1.2) we assume that there is an outcome space $\mathcal{Y} = \{r_1, \ldots, r_q\}$ with ordered ranks $r_q \succ_{\mathcal{Y}} r_{q-1} \succ_{\mathcal{Y}} \cdots \succ_{\mathcal{Y}} r_1$. The symbol $\succ_{\mathcal{Y}}$ denotes the ordering between different ranks and can be interpreted as "is preferred to." Since $\mathcal{Y}$ contains only a finite number of ranks, $P(y = r_i|\mathbf{x})$ is a multinomial distribution.

stochastic ordering

Let us make the assumption of stochastic ordering of the related space $\mathcal{X}$, i.e., for all different $\mathbf{x}_1$ and $\mathbf{x}_2$ either

$$\Pr(y \le r_i|\mathbf{x}_1) \ge \Pr(y \le r_i|\mathbf{x}_2) \qquad \text{for all } r_i \in \mathcal{Y}\,, \tag{7.1}$$

or

$$\Pr(y \le r_i|\mathbf{x}_1) \le \Pr(y \le r_i|\mathbf{x}_2) \qquad \text{for all } r_i \in \mathcal{Y}\,. \tag{7.2}$$

Stochastic ordering is satisfied by a model of the form

$$l^{-1}(\Pr(y \le r_i|\mathbf{x})) = \theta(r_i) - (\mathbf{w} \cdot \mathbf{x})\,, \tag{7.3}$$

model	inverse link function $P_\epsilon^{-1}(\Delta)$	density $dP_\epsilon(\eta)/d\eta$
logit	$\ln \frac{\Delta}{1-\Delta}$	$\frac{\exp(\eta)}{(1+\exp(\eta))^2}$
probit	$N^{-1}(\Delta)$	$\frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{\eta^2}{2}\right\}$
complementary log–log	$\ln(-\ln(1-\Delta))$	$\exp\{\eta - \exp(\eta)\}$

Table 7.1 Inverse link functions for different models for ordinal regression (taken from McCullagh and Nelder [1983]). Here, N^{-1} denotes the inverse normal function.

where $l^{-1} : [0,1] \mapsto (-\infty, +\infty)$ is a monotonic function often referred to as the inverse link function and $\theta : \mathcal{Y} \mapsto \mathbb{R}$ is increasing for increasing ranks. The stochastic ordering follows from the fact that

$$\begin{aligned}\Pr(y \le r_i|\mathbf{x}_1) \ge \Pr(y \le r_i|\mathbf{x}_2) &\Leftrightarrow \Pr(y \le r_i|\mathbf{x}_1) - \Pr(y \le r_i|\mathbf{x}_2) \ge 0 \\ &\Leftrightarrow l^{-1}(\Pr(y \le r_i|\mathbf{x}_1)) - l^{-1}(\Pr(y \le r_i|\mathbf{x}_2)) \ge 0 \\ &\Leftrightarrow (\mathbf{w} \cdot (\mathbf{x}_2 - \mathbf{x}_1)) \ge 0\,,\end{aligned}$$

which no longer depends on r_i (the same applies to $\Pr(y \le r_i|\mathbf{x}_1) \le \Pr(y \le r_i|\mathbf{x}_2)$).

cumulative model

Such a model is called a cumulative or threshold model and can be motivated by the following argument: Let us assume that the ordinal response is a coarsely measured *latent* continuous variable $U(\mathbf{x})$. Thus, we observe rank r_i in the training set iff

$$y = r_i \Leftrightarrow U(\mathbf{x}) \in [\theta(r_{i-1}), \theta(r_i)]\,, \tag{7.4}$$

where the function U (latent utility) and $\boldsymbol{\theta} = (\theta(r_0), \ldots, \theta(r_q))^T$ are to be determined from the data. By definition $\theta(r_0) = -\infty$ and $\theta(r_q) = +\infty$. We see that the real line is divided into q consecutive intervals, where each interval corresponds to a rank r_i. Let us make a linear model of the latent variable $U(\mathbf{x})$

linear utility model

$$U(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + \epsilon\,, \tag{7.5}$$

where ϵ is the random component of zero expectation, $\mathbf{E}_\epsilon(\epsilon) = 0$, and distributed according to P_ϵ. It follows from Equation (7.4) that

$$\begin{aligned}\Pr(y \le r_i|\mathbf{x}) &= \sum_{j=1}^{i} \Pr(y = r_j|\mathbf{x}) = \sum_{j=1}^{i} \Pr(U(\mathbf{x}) \in [\theta(r_{j-1}), \theta(r_j)]) \\ &= \Pr(U(\mathbf{x}) \in [-\infty, \theta(r_i)]) = \Pr((\mathbf{w} \cdot \mathbf{x}) + \epsilon \le \theta(r_i)) \\ &= P(\epsilon \le \underbrace{\theta(r_i) - (\mathbf{w} \cdot \mathbf{x})}_{\eta}) = P_\epsilon(\theta(r_i) - (\mathbf{w} \cdot \mathbf{x}))\,.\end{aligned}$$

If we now make a distributional assumption P_ϵ for ϵ we obtain the cumulative model by choosing as the inverse link function l^{-1} the inverse distribution function P_ϵ^{-1} (quantile function). Note that each quantile function $P_\epsilon^{-1} : [0,1] \mapsto (-\infty, +\infty)$ is a monotonic function. Different distributional assumptions for ϵ yield the logit, probit, or complementary log–log model (see Table 7.1).

In order to estimate $\mathbf{w}$ and $\boldsymbol{\theta}$ from model (7.3), for the observation $(\mathbf{x}_i, y)$ we see

$$\underbrace{\begin{pmatrix} o_1(\mathbf{x}_i) \\ o_2(\mathbf{x}_i) \\ \vdots \\ o_{q-2}(\mathbf{x}_i) \\ o_{q-1}(\mathbf{x}_i) \end{pmatrix}}_{\mathbf{o}(\mathbf{x}_i)} = \underbrace{\begin{pmatrix} -\mathbf{x}_i & 1 & 0 & \cdots & 0 & 0 \\ -\mathbf{x}_i & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ -\mathbf{x}_i & 0 & 0 & \cdots & 1 & 0 \\ -\mathbf{x}_i & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}}_{\mathbf{Z}(\mathbf{x}_i)} \underbrace{\begin{pmatrix} \mathbf{w} \\ \theta(r_1) \\ \theta(r_2) \\ \vdots \\ \theta(r_{q-2}) \\ \theta(r_{q-1}) \end{pmatrix}}_{\mathbf{w}_{\mathrm{GLM}}},$$

design matrix

where $o_j(\mathbf{x}_i) = P_\epsilon^{-1}(\Pr(y \le r_j|\mathbf{x}_i))$ is the transformed probability of ranks less than or equal to r_j given $\mathbf{x}_i$, which will be estimated from the sample by the transformed frequencies of that event. Note that the complexity of the model is determined by the linearity assumption (7.5) and by P_ϵ^{-1} which can be thought of as a regularizer in the resulting likelihood equation. For the complete training set we obtain

$$\underbrace{\begin{pmatrix} \mathbf{o}(\mathbf{x}_1) \\ \vdots \\ \mathbf{o}(\mathbf{x}_\ell) \end{pmatrix}}_{l^{-1}(\mathbf{y})\ \text{(random)}} = \underbrace{\begin{pmatrix} \mathbf{Z}(\mathbf{x}_1) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{Z}(\mathbf{x}_\ell) \end{pmatrix}}_{\mathbf{Z}\ \text{(random)}} \underbrace{\begin{pmatrix} \mathbf{w}_{\mathrm{GLM}} \\ \vdots \\ \mathbf{w}_{\mathrm{GLM}} \end{pmatrix}}_{\mathbf{W}_{\mathrm{GLM}}\ \text{(parameters)}} . \tag{7.6}$$

The last equation is called the design matrix of a multivariate generalized linear model (GLM). A generalized linear model $\mathbf{y} = l(\mathbf{Z}\mathbf{W}_{\mathrm{GLM}})$ is mainly determined by the design matrix $\mathbf{Z}$ and the link function $l(\cdot) = P_\epsilon(\cdot)$. Then given a sample (X, Y) and a link function — which coincides with a distributional assumption about the data — methods for calculating the maximum likelihood estimate $\mathbf{W}_{\mathrm{GLM}}$ exist (see McCullagh and Nelder [1983] or Fahrmeir and Tutz [1994] for a detailed discussion). The main difficulty in maximizing the likelihood is introduced by the nonlinear link function.

maximum likelihood estimate

To conclude this review of classical statistical methods we want to highlight the two main assumptions made for ordinal regression: (i) the assumption of stochastic ordering of the space $\mathcal{X}$ (ii) and a distributional assumption on the unobservable latent variable.

7.3 A Risk Formulation for Ordinal Regression

Instead of the distributional assumptions made in the last section, we now consider a parameterized model space G of mappings from objects to ranks. Each such function g induces an ordering $\succ_\mathcal{X}$ on the elements of the input space by the following rule

$$\mathbf{x}_i \succ_\mathcal{X} \mathbf{x}_j \Leftrightarrow g(\mathbf{x}_i) \succ_\mathcal{Y} g(\mathbf{x}_j) . \tag{7.7}$$

If we neglect the ordering of the space $\mathcal{Y}$, it was already shown in Section 1.1.1 that the Bayes–optimal function g^*_{class} given by Equation (1.5) is known to minimize

$$R_{\text{class}}(g) = \mathbf{E}_{\mathbf{x},y}\left(1_{g(\mathbf{x})\neq y}\right) = \mathbf{E}_{\mathbf{x},y}\left(c_{\text{class}}(\mathbf{x}, y, g(\mathbf{x}))\right) . \tag{7.8}$$

Let us rewrite $R_{\text{class}}(g)$ by

$$R_{\text{class}}(g) = \int_{\mathcal{X}} \mathcal{Q}_{\text{class}}(\mathbf{x}, g)\, p(\mathbf{x})d\mathbf{x}\,,$$

where

$$\mathcal{Q}_{\text{class}}(\mathbf{x}, g) = \sum_{i=1}^{q} \Pr(r_i|\mathbf{x}) - \Pr(g(\mathbf{x})|\mathbf{x}) = 1 - \Pr(g(\mathbf{x})|\mathbf{x})\,. \tag{7.9}$$

A closer look at Equation (7.9) shows that a sufficient condition for two mappings g_1 and g_2 to incur equal risks $R_{\text{class}}(g_1)$ and $R_{\text{class}}(g_2)$ is given by $\Pr(g_1(\mathbf{x})|\mathbf{x}) = \Pr(g_2(\mathbf{x})|\mathbf{x})$ for every $\mathbf{x}$. Assuming that $\Pr(r_i|\mathbf{x})$ is one for every $\mathbf{x}$ at a certain rank r_k the risks are equal — independently of how "far away" (in terms of rank difference) the mappings $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are from the optimal rank $\operatorname{argmax}_{r_i\in\mathcal{Y}} \Pr(r_i|\mathbf{x})$. This evidently shows that c_{class} is inappropriate for the case where a natural ordering is defined on the elements of $\mathcal{Y}$.

Since the only available information given by the ranks is the induced ordering of the input space $\mathcal{X}$ (see Equation (7.7)) we argue that a distribution independent model of ordinal regression has to single out that function g^*_{pref} which induces the ordering of the space $\mathcal{X}$ that incurs the smallest number of inversions on pairs $(\mathbf{x}_1, \mathbf{x}_2)$ of objects (for a similar reasoning see Sobel [1993]). To model this property we note that due to the ordering of the space $\mathcal{Y}$, each mapping g induces an ordering on the space $\mathcal{X}$ by Equation (7.7). Let use define the rank difference $\ominus : \mathcal{Y}\times\mathcal{Y} \mapsto \mathbb{Z}$ by

$$r_i \ominus r_j := i - j\,. \tag{7.10}$$

Now given a pair $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$ of objects we distinguish between two different events: $y_1\ominus y_2 > 0$ and $y_1\ominus y_2 < 0$. According to Equation (7.7) a function g violates the ordering if $y_1\ominus y_2 > 0$ and $g(\mathbf{x}_1)\ominus g(\mathbf{x}_2) \leq 0$, or $y_1\ominus y_2 < 0$ and $g(\mathbf{x}_1)\ominus g(\mathbf{x}_2) \geq 0$. Additionally taking into account that each weak order $\succ_{\mathcal{Y}}$ induces an equivalence $\sim_{\mathcal{Y}}$ [Fishburn, 1985] the case $y_1 \ominus y_2 = 0$ is automatically taken care of. Thus, an appropriate loss function is given by

loss function for ordinal regression

$$c_{\text{pref}}(\mathbf{x}_1, \mathbf{x}_2, y_1, y_2, g(\mathbf{x}_1), g(\mathbf{x}_2)) = \begin{cases} 1 & y_1 \ominus y_2 > 0 \wedge g(\mathbf{x}_1) \ominus g(\mathbf{x}_2) \leq 0 \\ 1 & y_2 \ominus y_1 > 0 \wedge g(\mathbf{x}_2) \ominus g(\mathbf{x}_1) \leq 0 \\ 0 & \text{else} \end{cases} \tag{7.11}$$

Note, that we can obtain m^2 samples drawn according to $p(\mathbf{x}_1, \mathbf{x}_2, y_1, y_2)$. It is important that these samples *do not provide* m^2 iid samples of the function $c_{\text{pref}}(\mathbf{x}_1, \mathbf{x}_2, y_1, y_2, g(\mathbf{x}_1), g(\mathbf{x}_2))$ for any g. Furthermore, if we define

$$c_g(\mathbf{x}_1, y_1, g(\mathbf{x}_1)) = \mathbf{E}_{\mathbf{x},y}\left[c_{\text{pref}}(\mathbf{x}_1, \mathbf{x}, y_1, y, g(\mathbf{x}_1), g(\mathbf{x}))\right]\,, \tag{7.12}$$

risk functional for ordinal regression

the risk functional to be minimized is given by

$$\begin{aligned} R_{\text{pref}}(g) &= \mathbf{E}_{\mathbf{x}_1,y_1,\mathbf{x}_2,y_2}\left(c_{\text{pref}}(\mathbf{x}_1,\mathbf{x}_2,y_1,y_2,g(\mathbf{x}_1),g(\mathbf{x}_2))\right) \\ &= \mathbf{E}_{\mathbf{x}_1,y_1}\left(c_g(\mathbf{x}_1,y_1,g(\mathbf{x}_1))\right) . \end{aligned} \tag{7.13}$$

Although Equation (7.13) shows great similarity to the classification learning risk functional (7.8) we see that due to the loss function c_g, which exploits the ordinal nature of $\mathcal{Y}$, we have a different pointwise loss function for each g . Thus we have found a risk functional which can be used for ordinal regression and takes into account the ordering as proposed by McCullagh and Nelder [1983].

In order to relate $R_{\text{pref}}(g)$ to a simple classification risk we slightly redefine the empirical risk based on c_{pref} and the training data (X,Y). For notational simplification let us define the space $\mathcal{E}$ of events of pairs $\mathbf{x}$ and y with unequal ranks by

$$\mathcal{E} := \{(\mathbf{z},t) \mid \mathbf{z}=(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{X}\times\mathcal{X}, t=\Omega(y_k,y_l), y_k\in\mathcal{Y}, y_l\in\mathcal{Y}, |y_k\ominus y_l|>0\}$$

Furthermore, using the shorthand notation $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ to denote the first and second object of a pair a new training set (X',Y') can be derived from (X,Y) if we use all 2–sets in $\mathcal{E}$ derivable from (X,Y), i.e.,

$$\forall\;\; 0<|y_i^{(1)}-y_i^{(2)}| \qquad (X',Y') = \left\{\left(\left(\mathbf{x}_i^{(1)},\mathbf{x}_i^{(2)}\right),\Omega\left(y_i^{(1)},y_i^{(2)}\right)\right)\right\}_{i=1}^{m'} \tag{7.14}$$

$$\Omega(y_1,y_2) := \operatorname{sgn}(y_1\ominus y_2)\,, \tag{7.15}$$

where Ω is an indicator function for rank differences and m' is the cardinality of (X',Y').

preference learning ⇔ classification

Theorem 7.1 Equivalence of Risk Functionals
Assume an unknown probability measure $p(\mathbf{x},y)$ on $\mathcal{X}\times\mathcal{Y}$ is given. Then for each $g:\mathcal{X}\mapsto\mathcal{Y}$ the following equalities hold true

$$R_{\text{pref}}(g) = \mathbf{E}_{y_1,y_2}\left(|\Omega(y_1,y_2)|\right)\mathbf{E}_{\mathbf{z},t}\left(c_{\text{class}}(\mathbf{z},t,\Omega(g(\mathbf{x}_1),g(\mathbf{x}_2)))\right)\,, \tag{7.16}$$

$$R_{\text{emp}}(g) = \frac{m'}{m^2}\sum_{i=1}^{m'} c_{\text{class}}\left(\left(\mathbf{x}_i^{(1)},\mathbf{x}_i^{(2)}\right),\Omega\left(y_i^{(1)},y_i^{(2)}\right),\Omega\left(g\left(\mathbf{x}_i^{(1)}\right),g\left(\mathbf{x}_i^{(2)}\right)\right)\right).$$

Proof Let us derive the probability $p(\mathbf{z},t)$ on $\mathcal{E}$ derived from $p(\mathbf{x}_1,\mathbf{x}_2,y_1,y_2)$:

$$p(\mathbf{z},t) = \begin{cases} 0 & t=0 \\ p(\mathbf{x}_1,\mathbf{x}_2,y_1,y_2)/\Delta & t\neq 0 \end{cases},$$

where

$$\Delta = \mathbf{E}_{y_1,y_2}\left(|\Omega(y_1,y_2)|\right) = \Pr(|y_1\ominus y_2|>0)\,.$$

Now exploiting the definition (7.11) of c_{pref} we see

$$\forall \mathbf{x}_1,\mathbf{x}_2,y_1,y_2,g : t = c_{\text{pref}}(\mathbf{x}_1,\mathbf{x}_2,y_1,y_2,g(\mathbf{x}_1),g(\mathbf{x}_2))\,.$$

The first statement is proven. The second statement follows by setting $\mathcal{X}=X,\mathcal{Y}=Y$ and assigning constant mass of $1/m^2$ at each point $(\mathbf{x}_1,\mathbf{x}_2,y_1,y_2)$. ■

Taking into account that each function $g \in G$ defines a function $p_g : \mathcal{X} \times \mathcal{X} \mapsto \{-1, 0, +1\}$ by

$$p_g(\mathbf{x}_1, \mathbf{x}_2) := \Omega(g(\mathbf{x}_1), g(\mathbf{x}_2)), \tag{7.17}$$

reduction to classification problem

Theorem 7.1 states that the empirical risk of a certain mapping g on a sample (X, Y) is equivalent to the c_{class} loss of the related mapping p_g on the sample (X', Y') up to a constant factor m'/m^2 which depends neither on g nor on p_g. Thus, the problem of distribution independent ordinal regression can be reduced to a classification problem on pairs of objects. It is important to emphasize the chain of argument that lead to this equivalence. The original problem was to find a function g that maps objects to ranks given a sample (X, Y). Taking the ordinal nature of ranks into account leads to the equivalent formulation of finding a function p_g that maps pairs of objects to the three classes $\succ_\mathcal{Y}$, $\prec_\mathcal{Y}$, and $\sim_\mathcal{Y}$. Reverting the chain of argumentation may lead to difficulties by observing that only those p_g are admissible — in the sense that there is a function g that fulfills Equation (7.17) — which define an asymmetric, transitive relation on $\mathcal{X}$. Therefore we also call this the problem of *preference learning*. It was shown that the Bayes optimal decision function given by (1.5) on pairs of objects can result in a function p_g which is no longer transitive on $\mathcal{X}$ [Herbrich et al., 1998]. This is also known as the problem of stochastic transitivity [Suppes et al., 1989]. Note also that the conditions of transitivity and asymmetry effectively reduce the space of admissible classification functions p_g acting on pairs of objects.

uniform convergence bounds

However, the above formulation is — in the form presented — not amenable to the straightforward application of classical results from learning theory. The reason is that the constructed samples of pairs of objects violate the iid assumption. In order to still be able to give upper bounds on a risk for preference learning we have to reduce our sample such that the resulting realization of the loss (7.11) is distributed iid. Under this condition it is then possible to bound the deviation of the expected risk from the empirical risk. Let σ be any permutation of the numbers $1, \ldots, m$. Furthermore, for notational convenience let $\mathcal{C}_g(i,j)$ abbreviate $c_{\text{pref}}(\mathbf{x}_i, \mathbf{x}_j, y_i, y_j, g(\mathbf{x}_i), g(\mathbf{x}_j))$. Then we see that for any $g \in G$

$$\begin{aligned} &\Pr(\mathcal{C}_g(\sigma(1),\sigma(2)), \mathcal{C}_g(\sigma(2),\sigma(3)), \ldots, \mathcal{C}_g(\sigma(m-1),\sigma(m))) \\ &= \Pr(\mathcal{C}_g(\sigma(1),\sigma(2))) \cdot \Pr(\mathcal{C}_g(\sigma(2),\sigma(3))) \cdot \ldots \cdot \Pr(\mathcal{C}_g(\sigma(m-1),\sigma(m))) . \end{aligned} \tag{7.18}$$

Clearly, $m-1$ is the maximum number of pairs of objects that still fulfil the iid assumption. In order to see this consider that by transitivity the ordering $g(\mathbf{x}_1) \prec_\mathcal{Y} g(\mathbf{x}_2)$ and $g(\mathbf{x}_2) \prec_\mathcal{Y} g(\mathbf{x}_3)$ implies $g(\mathbf{x}_1) \prec_\mathcal{Y} g(\mathbf{x}_3)$ (and vice versa for $\succ_\mathcal{Y}$ and $\sim_\mathcal{Y}$). Now we can give the following theorem.

Theorem 7.2 A Margin Bound for Ordinal Regression
Let p be a probability measure on $\mathcal{X} \times \{r_1, \ldots, r_q\}$, let (X, Y) be a sample of size m drawn iid from p. Let σ be any permutation of the numbers $1, \ldots, m$. For each function $g : \mathcal{X} \mapsto \{r_1, \ldots, r_q\}$ there exists a function $f \in F$ and a vector $\boldsymbol{\theta}$ such

that[1]

$$g(\mathbf{x}) = r_i \Leftrightarrow f(\mathbf{x}) \in [\theta(r_{i-1}), \theta(r_i)]\,. \tag{7.19}$$

Let the fat–shattering dimension of the set of functions F be bounded above by the function $\mathrm{afat}_F : \mathbb{R} \mapsto \mathbb{N}$. Then for each function g with zero training error, i.e., $\sum_{i=1}^{m-1} C_g(\sigma(i), \sigma(i+1)) = 0$ and

$$\rho_f = \min_{i=1,\ldots,m-1} \Omega\left(y_{\sigma(i)}, y_{\sigma(i+1)}\right) |f(\mathbf{x}_{\sigma(i)}) - f(\mathbf{x}_{\sigma(i+1)})|$$

with probability $1 - \delta$

$$R_{\mathrm{pref}}(g) \le \frac{2}{m-1}\left(k \log_2\left(\frac{8e(m-1)}{k}\right) \log_2(32(m-1)) + \log_2\left(\frac{8(m-1)}{\delta}\right)\right),$$

where $k = \mathrm{afat}_F(\rho_f/8) \le e(m-1)$.

Proof Let us recall the following theorem based on Theorem 1.10.

Theorem 7.3 [Shawe-Taylor et al., 1998]
Consider a real valued function class F having fat shattering function bounded above by the a function $\mathrm{afat}_F : \mathbb{R} \mapsto \mathbb{N}$ which is continuous from the right. Fix $\theta \in \mathbb{R}$. Then with probability $1 - \delta$ a learner that correctly classifies m iid generated examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ with $h = T_\theta(f) \in T_\theta(F)$ such that $h(\mathbf{x}_i) = y_i, i = 1, \ldots, m$ and $\rho_f = \min_i y_i (|f(\mathbf{x}_i) - \theta|)$ will have error of h bounded from above by

$$\frac{2}{m}\left(k \log_2\left(\frac{8em}{k}\right) \log_2(32m) + \log_2\left(\frac{8m}{\delta}\right)\right), \tag{7.20}$$

where $k = \mathrm{afat}_F(\rho_f/8) \le em$.

Taking into account that by construction we got $m - 1$ iid examples and that the classification of a pair is carried out by a decision based on the difference $f(\mathbf{x}_{\sigma(i)}) - f(\mathbf{x}_{\sigma(i+1)})$ we can upper bound $R_{\mathrm{pref}}(g)$ by replacing each m with $m - 1$ and using $\theta = 0$. ■

The $\mathrm{afat}_F(\rho)$–shattering dimension of F can be thought of as the maximum number of objects that can be arranged in any order using functions from F and a minimum margin $\min \Omega(y_1, y_2)|f(\mathbf{x}_1) - f(\mathbf{x}_2)|$ of ρ (utilizing Equation (7.7) together with (7.19)). Note, that the zero training error condition for the above bound is automatically satisfied for any σ if $R_{\mathrm{emp}}(g) = 0$. Even though this empirical risk was not based on an iid sample its minimization allows the application of the above bound. In the following section we will present an algorithm which aims at minimizing exactly that empirical risk while at the same time enforcing large margin rank boundaries.

1. Note the close relationship to the cumulative model presented in Section 7.2.

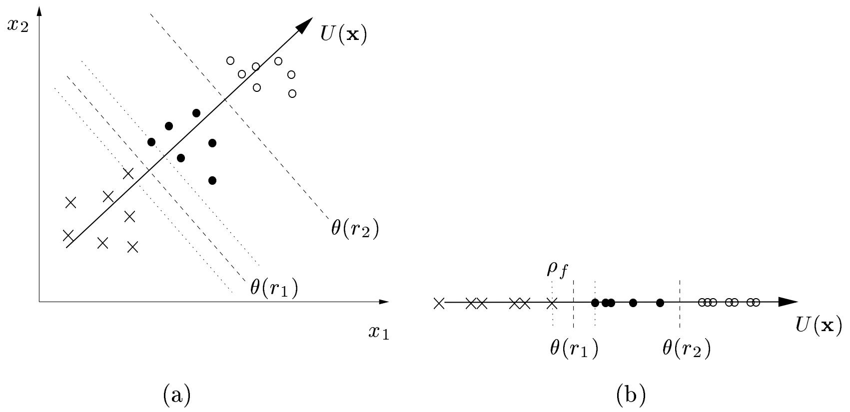


Figure 7.1 (**a**) Mapping of objects from rank r_1 ($\times$), rank r_2 ($\bullet$), and rank r_3 ($\circ$) to the axis $f(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2)^T$. Note that by $\theta(r_1)$ and $\theta(r_2)$ two coupled hyperplanes are defined. (**b**) The margin of the coupled hyperplanes $\rho_f = \min_{(X',Y')} \Omega(y_i^{(1)}, y_i^{(2)})|f(\mathbf{x}_i^{(1)}) - f(\mathbf{x}_i^{(2)})|$ is this time defined at the rank boundaries $\theta(r_i)$.

7.4 An Algorithm for Ordinal Regression

Based on the results of Theorem 7.2 we suggest to model ranks as intervals on the real line. Similarly to the classical cumulative model used in ordinal regression, let us introduce a (latent) linear function $f : \mathcal{X} \mapsto \mathbb{R}$ for each function g

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) , \tag{7.21}$$

ranks as intervals on the real line

which are related by (7.19). In order to apply the given theorem we see that we have to find a function f^* which incurs no training error on (X', Y') while controlling the generalization error by maximizing the margin ρ_f. Note, that

$$f(\mathbf{x}_i) - f(\mathbf{x}_j) = (\mathbf{w} \cdot (\mathbf{x}_i - \mathbf{x}_j)) ,$$

which makes apparent that each pair $(\mathbf{x}_i, \mathbf{x}_j) \in X'$ is represented by its difference vector $(\mathbf{x}_i - \mathbf{x}_j)$ assuming a linear model of f. This allows the straightforward application of the large margin algorithm given by Equation (1.51) and (1.52) replacing each $\mathbf{x}_i$ by $(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)})$. Hence, the maximization of the margin takes place at the rank boundaries $\theta(r_i)$ (see Equation (7.19) and Figure 7.1). In practice it is preferable to use the soft margin extension of the large margin algorithm (see Equation (1.25)). Furthermore due to the KKT conditions (see Equation (1.54)) $\mathbf{w}^*$ can be written in terms of the training data. This gives

$$\mathbf{w}^* = \sum_{i=1}^{m'} \alpha_i^* t_i \left(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)} \right) , \tag{7.22}$$

soft margin

where $\boldsymbol{\alpha}^*$ is given by

$$\boldsymbol{\alpha}^* = \operatorname*{argmax}_{\substack{C1 \geq \boldsymbol{\alpha} \geq 0 \\ (\boldsymbol{\alpha} \cdot \mathbf{t}) = 0}} \left[\sum_{i=1}^{m'} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m'} \alpha_i \alpha_j t_i t_j \left((\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}) \cdot (\mathbf{x}_j^{(1)} - \mathbf{x}_j^{(2)}) \right) \right], \tag{7.23}$$

and $\mathbf{t} = (\Omega(y_1^{(1)}, y_1^{(2)}), \ldots, \Omega(y_{m'}^{(1)}, y_{m'}^{(2)}))$. Note, however, that due to the expansion of the last term in (7.23),

$$\left((\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}) \cdot (\mathbf{x}_j^{(1)} - \mathbf{x}_j^{(2)}) \right) = (\mathbf{x}_i^{(1)} \cdot \mathbf{x}_j^{(1)}) - (\mathbf{x}_i^{(1)} \cdot \mathbf{x}_j^{(2)}) - (\mathbf{x}_i^{(2)} \cdot \mathbf{x}_j^{(1)}) + (\mathbf{x}_i^{(2)} \cdot \mathbf{x}_j^{(2)}),$$

the solution $\boldsymbol{\alpha}^*$ to this problem can be calculated solely in terms of the inner products between the feature vectors without reference to the feature vectors themselves. Hence, the idea of (implicitly) mapping the data X via a nonlinear mapping $\Phi : \mathcal{X} \mapsto \mathcal{F}$ into a feature space $\mathcal{F}$ can successfully applied (for further details see Section 1.3.2). Replacing each occurrence of $\mathbf{x}$ by $\Phi(\mathbf{x})$ gives

kernel trick

$$\boldsymbol{\alpha}^* = \operatorname*{argmax}_{\substack{C1 \geq \boldsymbol{\alpha} \geq 0 \\ (\boldsymbol{\alpha} \cdot \mathbf{t}) = 0}} \left[\sum_{i=1}^{t} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{t} \alpha_i \alpha_j t_i t_j \mathcal{K} \left(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \mathbf{x}_j^{(1)}, \mathbf{x}_j^{(2)} \right) \right]. \tag{7.24}$$

where $\mathcal{K}$ is for a given function k defined by

$$\mathcal{K}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = k(\mathbf{x}_1, \mathbf{x}_3) - k(\mathbf{x}_1, \mathbf{x}_4) - k(\mathbf{x}_2, \mathbf{x}_3) + k(\mathbf{x}_2, \mathbf{x}_4). \tag{7.25}$$

Here, $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a Mercer kernel and for a fixed mapping Φ is defined by

$$k(\mathbf{x}, \mathbf{x}') = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')).$$

Some kernels k to be used in learning are given by Equations (1.63) and (1.73). Note that the usage of kernels instead of explicitly performing the mapping Φ allows us to deal with nonlinear functions f without running into computational difficulties. Moreover, as stated in Theorem 7.2 the bound on the risk $R_{\text{pref}}(\mathbf{w})$ does not depend on the dimension of $\mathcal{F}$ but on the margin ρ_f.

In order to estimate the rank boundaries we note that due to Equations (1.52) the difference in f^* is greater or equal to one for all training examples which constitute a correctly classified pair. These can easily be obtained by checking $0 < \alpha_i^* < C$, i.e., training patterns which do not meet the box constraint (see Section 1.1.4). Thus if $\Theta(k) \subset X'$ is the fraction of objects from the training set with $0 < \alpha_i^* < C$ and rank difference $\ominus$ exactly one starting from rank r_k, i.e.,

rank boundaries

$$\Theta(k) = \left\{ \left(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)} \right) \middle| y_i^{(1)} = r_k \wedge y_i^{(2)} = r_{k+1} \wedge 0 < \alpha_i^* < C \right\} \tag{7.26}$$

then the estimation of $\theta(r_k)$ is given by

$$\theta^*(r_k) = \frac{f^*(\mathbf{x}_1) + f^*(\mathbf{x}_2)}{2}, \tag{7.27}$$

where

$$(\mathbf{x}_1, \mathbf{x}_2) = \operatorname*{argmin}_{(\mathbf{x}_i, \mathbf{x}_j) \in \Theta(k)} [f^*(\mathbf{x}_i) - f^*(\mathbf{x}_j)]. \tag{7.28}$$

In other words, the optimal threshold $\theta^*(r_k)$ for rank r_k lies in the middle of the utilities of the closest (in the sense of their utility) objects of rank r_k and r_{k+1}. After the estimation of the rank boundaries $\theta(r_k)$ a new object is assigned to a rank according to Equation (7.19).

coupled hyperplanes

We want to emphasize that taking the difference vector as a representation of a pair of objects effectively couples all hyperplanes $f(\mathbf{x}) = \theta(r_k)$ thus resulting in a standard QP problem. Furthermore, the effective coupling is retained if we use general ℓ_q–margins (see Section 1.1.4). It is the reduction of the hypothesis space which makes the presented algorithm suited for the task of ordinal regression. Note, that also the kernel $\mathcal{K}$ derived from k acts only in $\mathcal{F}$ and thus avoids considering too large a hypothesis space. All properties are consequences of the *modeling of ranks* as intervals on the real line and of the prior knowledge of the ordering of $\mathcal{Y}$.

7.5 Experimental Results

In this section we present some experimental results for the algorithm presented in Section 7.4. We start by giving results for artificial data which allows us to analyze our algorithm in a controlled setting. Then we give learning curves for an example from the field of information retrieval.

7.5.1 Learning Curves for Ordinal Regression

multi-class SVM and support vector regression

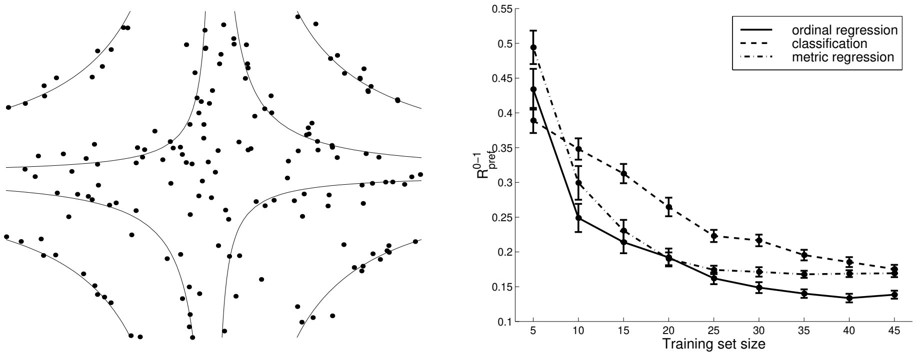
In this experiment we want to compare the generalization behavior of our algorithm with the multi-class SVM [Weston and Watkins, 1998] and Support Vector regression (SVR) (cf. Smola [1998]) — the methods of choice, if one does not pay attention to the ordinal nature of $\mathcal{Y}$ and instead treats ranks as classes (classification) or continuous response values (regression estimation). Another reason for choosing those algorithms is their similar regularizer $\|\mathbf{w}\|^2$ and hypothesis space F which make them as comparable as possible. We generated 1000 observations $\mathbf{x} = (x_1, x_2)$ in the unit square $[0,1] \times [0,1] \subset \mathbb{R}^2$ according to a uniform distribution. We assigned to each observation $\mathbf{x}$ a value y according to

$$y = i \Leftrightarrow \underbrace{10((x_1 - 0.5) \cdot (x_2 - 0.5))}_{f(\mathbf{x})} + \epsilon \in [\theta(r_{i-1}), \theta(r_i)]\,, \tag{7.29}$$

example utility function

where ϵ was normally distributed, i.e., $\epsilon \sim N(0, 0.125)$, and $\boldsymbol{\theta} = (-\infty, -1, -0.1, 0.25, 1, +\infty)$ is the vector of predefined thresholds. In Figure 7.2 (a) the points $\mathbf{x}_i$ which are assigned to a different rank after the addition of the normally distributed quantity ϵ_i are shown. If we treat the whole task as a classification problem, we would call them incorrectly classified training examples. The solid lines in Figure 7.2 (a) indicate the "true" rank boundaries $\boldsymbol{\theta}$ on $f(\mathbf{x})$.

In order to compare the three different algorithms we randomly drew 100 training samples (X, Y) of training set sizes m ranging from 5 to 45, thereby making sure



(a) (b)

Figure 7.2 (**a**) Scatter plot of data points $\mathbf{x}$ which $f(\mathbf{x})$ maps to a different interval than $f(\mathbf{x})+\epsilon$ (see Equation (7.29)). (**b**) Learning curves for multi-class SVM (dashed lines), SV regression (dashed–dotted line) and the algorithm for ordinal regression (solid line) if we measure R_{pref}. The error bars indicate the 95% confidence intervals of the estimated risk R_{pref}.

comparison to other methods

that at least one representative of each rank was within the drawn training set. Classification with multi-class SVMs was carried out by computing the pairwise $5 \cdot 4/2 = 10$ hyperplanes. For all algorithms, i.e., multi-class SVMs, SVR, and the algorithm presented in Section 7.4, we chose the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i \cdot \mathbf{x}_j) + 1)^2$ and a trade-off parameter $C = 1000000$. In the particular case of Support Vector regression we used a value of $\varepsilon = 0.5$ for the ε–insensitive loss function (see [Vapnik, 1995] for the definition of this loss function) and thresholds $\boldsymbol{\theta} = (0.5, 1.5, 2.5, 3.5, 4.5)$ to transform real valued predictions into ranks.

learning curves

In order to estimate the risk $R_{\text{pref}}(g^*)/\mathbf{E}_{y_1,y_2}(|\Omega(y_1, y_2)|)$ from the remaining 995 to 955 data points we averaged over all 100 results for a given training set size. Thus we obtained the three learning curves shown in Figure 7.2 (b). Note that we used the scaled R_{pref} — which is larger by a constant factor. It can be seen that the algorithm proposed for ordinal regression generalizes much faster by exploiting the ordinal nature underlying $\mathcal{Y}$ compared to classification. This can be explained by the fact that due to the model of a latent utility all "hyperplanes" $f(\mathbf{x}) = \theta(r_k)$ are coupled (see Figure 7.1) which does not hold true for the case of multi-class SVMs. Furthermore, the learning curves for SVR and the proposed ordinal regression algorithm are very close which can be explained by the fact that the predefined thresholds $\theta(r_k)$ are defined in such a way that their pairwise difference is about 0.5 — the size of the ε–tube chosen beforehand. Thus the utility and the continuous ranks estimated by the regression algorithm are of the same magnitude which results in the same generalization behavior.

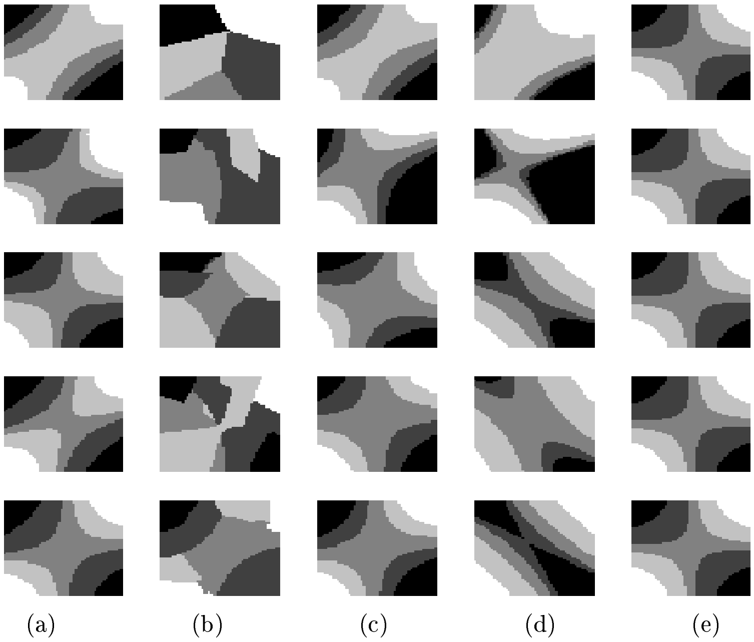
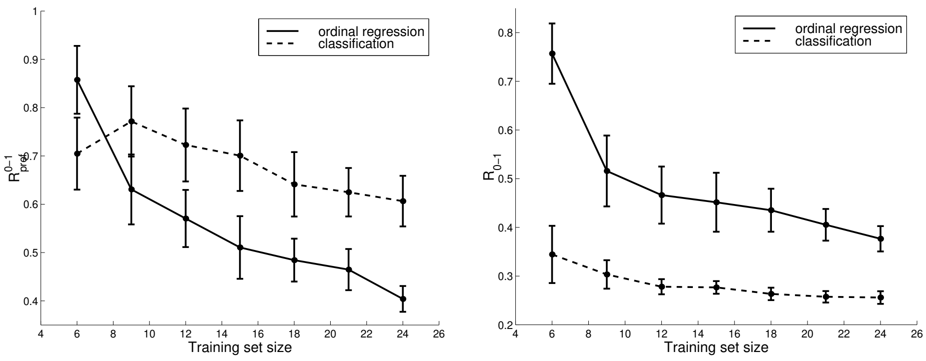


Figure 7.3 Assignments of points to ranks r_1 (black area) to r_5 (white area) by the learned function $g^*(\mathbf{x})$ based on randomly drawn training samples of size $5, 10, 15, 20$, and 25 (top row to bottom row). (**a**) Results of the algorithm presented in Section 7.4. (**b**) Results of multi-class SVM if we treat each rank as a class. (**c**) Results of SVR if we assign rank r_i to number i. (**d**) Results of SVR if we assign rank r_i to real number $\exp(i)$. (**e**) Underlying assignment uncorrupted by noise.

In Figure 7.3 we plotted the assignments of the unit square to ranks r_1 (black areas) to ranks r_5 (white areas) for the functions $g^*(\mathbf{x})$ learned from randomly drawn training sets ranging from size $m = 5$ (top row) to $m = 25$ (bottom row). We used the same parameters as for the computation of the learning curves. In the rightmost column (e) the true assignment, i.e., $y = r_i \Leftrightarrow f(\mathbf{x}) \in [\theta(r_{i-1}), \theta(r_i)]$ is shown. In the first column (a) we can see how the algorithm presented in Section 7.4 performs for varying training set sizes. As expected, for the training set size $m = 25$, the method found a utility function together with a set of thresholds which represent the true ranking very well. The second column (b) shows the results of the abovementioned multi-class SVM on the task. Here the pairwise hyperplanes are not coupled since the ordinal nature of $\mathcal{Y}$ is not taken into account. This results in a worse generalization, especially in regions, where no training points were given. The third column (c) gives the assignments made by the SVR algorithm if we represent each rank r_i by i. Similar to the good results seen in the learning curve, the generalization behavior is comparable to the ordinal regression method (first column). The deficiency of SVR for this task becomes apparent when we change the representation of ranks. In the fourth column (d) we applied the same SVR



(a) (b)

Figure 7.4 Learning curves for multi-class SVM (dashed lines) and the algorithm for ordinal regression (solid line) for the OHSUMED dataset query 1 if we measure (**a**) R_{pref} and (**b**) R_{class}. Error bars indicate the 95% confidence intervals.

representation of ranks

algorithm, this time on the representation $\exp(i)$ for rank r_i. As can be seen, this dramatically changes the generalization behavior of the SVR method. We conclude that the crucial task for application of metric regression estimation methods to the task of ordinal regression is the definition of the representation of ranks. This is automatically — although more time–consuming — solved by the proposed algorithm.

7.5.2 An Application to Information Retrieval

information retrieval

In this experiment we make the following assumption: After an initial (textual) query a user makes to an IR system, the system returns a bundle of documents to the user. Now the user assigns ranks to a small fraction of the returned documents and the task for the learning algorithm is to assign ranks to the remaining unranked documents in order to rank the remaining documents. We assume that the quantity of interest is the percentage of inversions incurred by the ranking induced by the learning algorithm. This quantity is captured by $R_{\text{emp}}(g)/m'$ ($m' = |(X', Y')|$, see Equation (7.14) for an exact definition) and thus after using $m = 6$ up to $m = 24$ documents and their respective ranking we measure this value on the remaining documents. For this experiment we used the same parameters as in the previous experiment. The investigated dataset was the OHSUMED dataset collected by William Hersh[2], which consists of 348 566 documents and 106 queries with their respective ranked results. There are three ranks: "document is relevant," "document is partially relevant," and "irrelevant document" wrt. the given textual

2. This dataset is publicly available at `ftp://medir.ohsu.edu/pub/ohsumed/`.

query. For our experiments we used the results of query 1 ("Are there adverse effects on lipids when progesterone is given with estrogen replacement therapy?") which consists of 107 documents taken from the whole database. In order to apply our algorithm we used the bag–of–words representation [Salton, 1968], i.e., we computed for every document the vector of "term–frequencies–inverse–document–frequencies" (TFIDF) components. The TFIDF is a weighting scheme for the bag–of–words representation which gives higher weights to terms which occur very rarely in all documents. We restricted ourselves to terms that appear at least three times in the whole database. This results in $\approx$ 1700 terms which leads for a certain document to a very high–dimensional but sparse vector. We normalized the length of each document vector to unity (see Joachims [1998]).

bag–of–words representation

Figure 7.4 (a) shows the learning curves for multi-class SVMs and our algorithm for ordinal regression measured in terms of the number of incurred inversions. As can be seen from the plot, the proposed algorithm shows very good generalization behavior compared to the algorithm which treats each rank as a separate class. Figure 7.4 (b) shows the learning curves for both algorithms if we measure the number of misclassifications — treating the ranks as classes. As expected, the multi-class SVMs perform much better than our algorithm. It is important to note again, that minimizing the zero–one loss R_{class} does not automatically lead to a minimal number of inversions and thus to an optimal ordering.

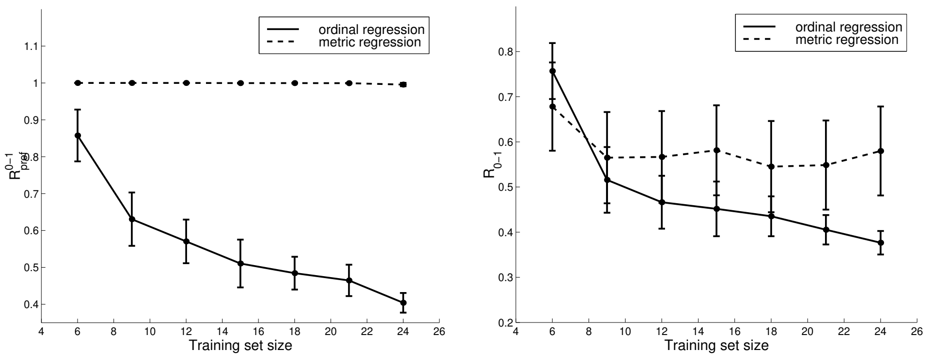


Figure 7.5 Learning curves for SVR (dashed lines) and the algorithm for ordinal regression (solid line) for the OHSUMED dataset query 1 if we measure (**a**) R_{pref} and (**b**) R_{0-1}. Error bars indicate the 95% confidence intervals.

Figure 7.5 (a) shows the learning curves for SVR and for our algorithm for ordinal regression, measured the number of incurred inversions. While the former performs quite well on the artificial dataset, in the real world dataset the SVR algorithm fails to find a ranking which minimizes the number of inversions. This can be explained by fact that for the real–world example the equidistance in the assumed utility may no longer hold — especially taking into account that the data space is very sparse for this type of problem. Similarly, Figure 7.5 (b) shows the learning curves for both algorithms if we measure the number of misclassifications. As expected from the curves on the right the SVR algorithm is worse even on that measure. Note that the SVR algorithm minimizes neither R_{pref} nor R_{0-1} which may explain its bad generalization behavior. Also note that we made no adaptation of the parameter ε — the size of the tube. The reason is that in this particular task there would not be enough training examples available to set aside a reasonable portion of them for validation purposes.

7.6 Discussion and Conclusion

In this chapter we considered the task of ordinal regression which is mainly characterized by the ordinal nature of the outcome space $\mathcal{Y}$. All known approaches to this problem (see Section 7.2) make distributional assumptions on an underlying continuous random variable. In contrast, we proposed a loss function which allows for application of distribution independent methods to solve ordinal regression problems. By exploiting the fact that the induced loss function class is a set of indicator functions we could give a distribution independent bound on our proposed risk. Moreover, we could show that to each ordinal regression problem there exists a corresponding preference learning problem on pairs of objects. This result built the link between ordinal regression and classification methods — this time on pairs of objects. For the representation of ranks by intervals on the real line, we could give margin bounds on our proposed risk — this time applied at the rank boundaries. Based on this result we presented an algorithm which is very similar to the well known Support Vector algorithm but effectively couples the hyperplanes used for rank determination.

learning of equivalence relation

Noting that our presented loss involves pairs of objects we see that the problem of multi-class classification can also be reformulated on pairs of objects which leads to the problem of learning an *equivalence relation*. Usually, in order to extend a binary classification method to multiple classes, one–against–one or one–against–all techniques are devised [Hastie and Tibshirani, 1998, Weston and Watkins, 1998]. Such techniques increase the size of the hypothesis space quadratically or linearly in the number of classes, respectively. Recent work [Phillips, 1999] has shown that learning equivalence relations can increase the generalization behavior of binary–class methods when extended to multiple classes.

Further investigations will include the following question: does the application of the GLM methods presented in Section 7.2 lead automatically to large margins (see Theorem 7.2)? The answer to such a question would finally close the gap between methods extensively used in the past to theories developed currently in the field of Machine Learning.

Acknowledgments

First of all we are indebted to our collaborator Peter Bollmann-Sdorra who first stimulated research on this topic. We also thank Nello Cristianini, Ulrich Kockelkorn, Gerhard Tutz, and Jason Weston for fruitful discussions. Finally, we would like to thank our anonymous reviewers for very helpful comments on the uniform convergence results.