Published in final edited form as:
IEEE Trans Neural Netw Learn Syst. 2016 October ; 27(10): 2072–2083. doi:10.1109/TNNLS.2015.2477321.

Multiple Ordinal Regression by Maximizing the Sum of Margins

Onur C. Hamsici and
Qualcomm Research, San Diego, CA

Aleix M. Martinez
Department of Electrical and Computer Engineering, The Ohio State University

Onur C. Hamsici: ohamsici@qti.qualcomm.com; Aleix M. Martinez: aleix@ece.osu.edu

Abstract

Human preferences are usually measured using ordinal variables. A system whose goal is to estimate the preferences of humans and their underlying decision mechanisms requires to learn the ordering of any given sample set. We consider the solution of this *ordinal regression* problem using a Support Vector Machine algorithm. Specifically, the goal is to learn a set of classifiers with common direction vectors and different biases correctly separating the *ordered* classes. Current algorithms are either required to solve a quadratic optimization problem, which is computationally expensive, or are based on maximizing the minimum margin (i.e., a *fixed margin strategy*) between a set of hyperplanes, which biases the solution to the closest margin. Another drawback of these strategies is that they are limited to order the classes using a *single* ranking variable (e.g., perceived length). In this paper, we define a *multiple* ordinal regression algorithm based on maximizing the sum of the margins between *every* consecutive class with respect to one or more rankings (e.g., perceived length and weight). We provide derivations of an efficient, easy-to-implement iterative solution using a Sequential Minimal Optimization procedure. We demonstrate the accuracy of our solutions in several datasets. In addition, we provide a key application of our algorithms in estimating human subjects' ordinal classification of attribute associations to object categories. We show that these ordinal associations perform better than the binary one typically employed in the literature.

Index Terms

Ordinal Regression; SVM; Sequential Minimal Optimization; Computer Vision; Object Classification; Direct Attribute Prediction

I. Introduction

Ordinal measurements play a key role in many applications, from plant biology [41] to users' reviews [43] and visual perception [45]. For instance, objects can be ordered according to their features, functionality, etc. A clear example is in the perception of facial expressions of emotion by humans. These facial expressions are generally ordered according to the degree of the perceived emotion category, regardless of the actual category being

---

expressed by the sender [27], [9], [8]. Unlike a classification problem, here, errors generated by misclassifications reflect the difference between the orderings, and unlike a regression problem, the labels define the discrete class ranks [16], [6]. In our facial expression example, the perceived difference (i.e., distance) of an emotion category shown in images *a* and *b* is given by the difference between two ordinal variables [27], [23]. Ordinal measurements only provide the order of the classes relative to each other.

In the present paper, we derive a *multiple* ordinal regression algorithm, where two or more ranking (unknown) functions are used to order the data. Our goal is to simultaneously estimate these multiple underlying ranking functions to minimize the global ordinal classification error. Existing ordinal classification algorithms described in the literature are either computationally complex or based on maximizing the number of correct pairwise rankings which is sensitive to outliers [13]. To resolve these problems, in the present paper, we derive a Support Vector Machine (SVM)-based formulation defined by a simple iterative approach which minimizes the global risk. We first present the single ranking solution and then derive the algorithm for multiple ranking functions. This allows us to weight the importance of the ranking problems with respect to each other. Finally, we derive an efficient, easy-to-solve iterative solution using a Sequential Minimal Optimization (SMO) based iterative procedure.

Extensive experimental results show the accuracy of the algorithm through several tests in the UCI datasets. Importantly, we illustrate an application of our algorithm in learning the visual feature rankings of human subjects, where we learn to rank objects based on user-specified, high-level descriptions (attributes). This allows developing detection and recognition algorithms that are based on high-level attribute ranking estimations provided by our algorithm. We illustrate this scheme in Fig. 1. In the experimental results section, we also provide an application of this multiple ranking algorithm to data visualization.

A. Background and significance

Several algorithms have been proposed to resolve the *ordinal regression* problem [26], [47], [12], [22], [1], [6], [38], [18], [13], [20], [17]. For example in [20] the authors learn a regression tree to estimate the ordinal values. The limitations of this approach are that the metric between the ordinal variables is not defined and that the regression algorithm assumes a uniform error.

To resolve the above stated problems, Shashua et al. [38] derive SVM-based ranking algorithms to address the *single* ranking problem. The goal of their algorithms is to learn a set of hyperplanes with common weight vector and different biases to order a set of ordinal classes. Shashua et al. present two possible solutions to this problem – a *fixed margin* strategy and a *sum of margins* strategy. Fig. 2 shows a typical solution of these two strategies. The fixed margin strategy maximizes the distance of the closest class pair. This solution is prone to errors, since the closest class pair may not define a correct direction of separation for all classes. In contrast, the second strategy maximizes the sum of the margins between every consecutive class while minimizing the global risk. Unfortunately, the formulation of [38] leads to several inequalities that are very difficult to solve and are inefficient. Both of these algorithms are in fact formulated as quadratic optimization problems and, hence, the number of samples that can be utilized is limited.

Chu et.al. [6] address the above limitations by reformulating the fixed margin approach as a SMO[1] (Sequential Minimal Optimization). In addition, [6] extended the fixed margin solution to handle the implicit errors that would be generated when a sample is misranked further than two classes away from its correct class rank. Unfortunately, these solutions were based on the fixed margin strategy and are hence biased towards the classification direction given by the closest class pairs.

Other types of ranking formulations have been commonly used in the information retrieval community for ranking document retrieval results [17], [18], [1], [22]. In these cases, the ranking SVM problem is formulated as to find a solution that maximizes the number of correctly ordered pairwise samples. Note that, this problem is different than our ordinal classification learning or ordinal regression problem. While we are learning a function to estimate an ordinal variable, these approaches target to maximize correctly ordered pairwise samples. Considering all pairwise samples, this results in a large quadratic optimization with complexity squared with the number of samples. To address this issue, several efficient implementations of ranking SVM with a fixed margin strategy have been proposed [19], [4], [3], [49]. Boosting approaches defined in the ”Yahoo! Learning to Rank Challenge” [3], such as GBRANK [49], use a similar formulation to ranking SVM. Efficient ranking learning with primal optimization techniques have been defined in [36]. A recent approach by [7] derives ranking forests to address learning to rank problem with binary feedbacks. Again, all these approaches are defined for the information retrieval applications and target to learn a ranking SVM that maximize the correct pairwise orderings and are hence based on a *fixed margin strategy*, with the subsequent limitations already stated above.

In a recent study, [44] provides a detailed theoretical comparison between various ranking algorithms. It is shown that most of these approaches are related to the proportional log odds model in statistics [25]. Specifically, the fixed-margin strategy based solution of [6] with the implicit constraints on individual samples have the same asymptotic ranking function as the proportional log odds model. However, [44] points out that the dependency of this solution to the implicit constraints of individual sample rankings may result in poor performance in some applications. This provides a theoretical support for our observations on the inferior performance of the fixed margin strategy. [44] also emphasizes that the objective functions for ordinal regression and multiclass classification are significantly different, and multiclass classification would result in large errors if used in an ordinal regression problem. This is also reflected in our experimental results in Section IV. These methods are also to be compared to recent algorithms that search for a balance between the bias and variance of the regressors [47]. The major problem of this approach is its limitations when working with multiple ordinal variables.

There has also been a great interest on the theoretical properties of the learning to rank approaches from partial pairwise rankings. [10] provides a detailed review of the learning to

---

[1] SMO is an iterative solution to a computationally complex quadratic optimization problem in SVM [31].

rank algorithms and summarizes that the practical solutions to collect ranking information don't have a relevant theory and the approaches that have theoretical support don't have practical solutions. To address these limitations, [10] proposes a theory to identify partial preferences when the dataset is only partially labeled. [33] also provides a solution for the rank aggregation from pairwise data and derives theoretical results for the underlying statistical model. Note that, we provide the review of these recent approaches to clarify that the ordinal regression problem we are addressing in this paper is a different problem from the approaches that are designed for learning to rank problem.

Other recent ordinal regression approaches provide extensions for ensemble learning [29], [11], sampling problems [30], and semi-supervised learning [37]. [29] proposes an ensemble learning for ordinal regression by extracting multiple projection-based two class classifiers and three class ordinal regressors. The ensemble of the probability scores obtained from these functions are used to rank the test samples. [11] addresses the feature selection for ensemble learning of ordinal regressors. Negative correlation is used to find new features that provide additional information to the ensemble. Both of these ensemble solutions are shown to improve the accuracy of fixed margin strategy based solutions [6], Gaussian Processes Ordinal Regression [6] and SVM. [30] addresses the sampling problems in ordinal regression using a graph-based approach. Ranking classes that have large number of samples usually dominate the regression and results in misranking of classes when the number of samples is small. This is addressed by a graph-based approach to properly adjust the regressor weights and is shown to improve the accuracy of fixed margin solutions. [37] proposes a semi-supervised approach of Transductive Ordinal Regression. Unlabeled samples are utilized while learning the regressor. A training algorithm is proposed to estimate the labels of the unlabeled samples and minimize the loss function for the samples with ranking labels. This approach could be utilized when the data labels are missing and it is difficult to collect the rankings for all classes. These extensions are all proposed for ordinal regression problems and can utilize our sum-of-margin based multiple and single ordinal regression solutions proposed in the present paper. This increases the applicability of our solutions to a larger set of problems.

II. Single Ranking Learning with SVM

Assume we are given a set of samples $\mathbf{x}_i^j \in \mathbb{R}^d$, where $i = \{1, 2, \ldots, n_j\}$ specifies the sample, n_j is the total number of samples in class j, and the superscript $j = \{1, 2, \ldots, R\}$ is the class label which specifies the *order* of the sample.

Our goal is to learn a ranking function that estimates the ordering of a future (test) observation. Let this function be represented by a set of parallel hyperplanes, with **w** as the common direction vector. Let each class lie between two of the resulting $R - 1$ hyperplanes with biases $b_1, b_2, \ldots, b_{R-1}$. We classify a test sample **x** to (class) rank r using the rule,

$$f(\mathbf{x}) = \min_{r \in 1,2,\ldots,R} r : \mathbf{w}^T \mathbf{x} < b_r, \quad (1)$$

Note that, without loss of generality, the rank R hyperplane is set at infinity with $b_R = \infty$.

Fig. 2(b) illustrates the geometric interpretation of the ranking problem. In this example, our goal is to estimate the ranking function that maximizes the separation of the samples in 3 classes (shown as circles, plus signs and squares in the figure) with 4 parallel hyperplanes. The vector **w** illustrates the common weight vector for the hyperplanes with biases a_1, b_1, a_2 and b_2. The margins between consecutive classes are defined as $b_i - a_i$. The averages of the a_i and b_i are used to define the classification rule, i.e., $c_i=\frac{a_i+b_i}{2}$. The classifier $\mathbf{w}^T\mathbf{x} \leq c_1$ specifies the ordering obtained by the first function, $c_1 < \mathbf{w}^T\mathbf{x} \leq c_2$ given by the second rank, and $\mathbf{w}^T\mathbf{x} > c_2$ of the third. For this problem, our ranking function is $f(x) = \min_{r=1,\ldots,R} \mathbf{w}^T\mathbf{x} < c_r$, with $c_R = \infty$ and $R = 3$. Note that, the hyperplanes $\mathbf{w}^T\mathbf{x} = a_1$ and $\mathbf{w}^T\mathbf{x} = a_2$ penalize the misranking errors of samples from classes 1 and 2 to classes with higher ranks, e.g., for samples from class 1 these are classes 2 and 3. Similarly, $\mathbf{w}^T\mathbf{x} = b_1$ and $\mathbf{w}^T\mathbf{x} = b_2$ penalize misorderings of samples from classes 2 and 3 to classes with lower rank. We now derive this geometric problem with a SVM type ranking function learning.

Formally, we have

$$\begin{aligned} &\min F\left(\mathbf{w},\varepsilon_i^j,\varepsilon_i^{*j+1},a_i,b_i|C\right)=\frac{1}{2}\|\mathbf{w}\|^2+\\ &C\sum_{j=1}^{R-1}\left(\sum_{i=1}^{n_j}\varepsilon_i^j+\sum_{i=1}^{n_{j+1}}\varepsilon_i^{*j+1}\right)+\sum_{j=1}^{R-1}(a_j-b_j)\\ &\text{such that}\begin{cases}\mathbf{w}^T\mathbf{x}_i^j-a_j\leq\varepsilon_i^j,-\mathbf{w}^T\mathbf{x}_i^{j+1}+b_j\leq\varepsilon_i^{*j+1},\\ \qquad\varepsilon_i^j\geq 0,\quad \varepsilon_i^{*j+1}\geq 0,\end{cases}\\ &\text{for } j=\{1,\ldots,R-1\},i=\{1,\ldots,n_j\}, \end{aligned} \qquad (2)$$

where ε_i^j and ε_i^{*j+1} are the slack variables that correspond to errors in the orderings if the samples $\mathbf{x}_i^j$ and $\mathbf{x}_i^{j+1}$ with respect to the hyperplanes with biases a_j and b_j, and where n_j is the number of samples in class j.

The first term in the optimization problem above corresponds to the complexity of the classifier and the common scale of the margins. The second term is the sum of errors generated by the hyperplanes, and the third term is the sum of margins between consecutive hyperplanes. The parameter C controls the importance of the error term with respect to the margin. The inequality conditions define the classification rule and the positivity of the slack variables ε.

The Lagrangian for the primal problem in (2) is given by,

$$\begin{gathered}L=\frac{1}{2}\|\mathbf{w}\|^2+C\left(\sum_{i,j}\varepsilon_i^j+\sum_{i,j}\varepsilon_i^{*j+1}\right)\\+\sum_{j=1}^{R-1}(a_j-b_j)+\sum_{i,j}\lambda_i^j\left(\mathbf{w}^T\mathbf{x}_i^j-a_j-\varepsilon_i^j\right)\\+\sum_{i,j}\delta_i^j\left(-\mathbf{w}^T\mathbf{x}_i^{j+1}+b_j-\varepsilon_i^{*j+1}\right)\\-\sum_{i,j}\gamma_i^j\varepsilon_i^j-\sum_{i,j}\gamma_i^{*j+1}\varepsilon_i^{*j+1}\end{gathered} \quad (3)$$

where $\lambda_i^j,\delta_i^j,\gamma_i^j,\gamma_i^{*j+1}$ are the nonnegative Lagrange multipliers.

The KKT (Karush-Kuhn-Tucker) optimality conditions for the above problem are,

$$\frac{\partial L}{\partial \mathbf{w}}=\mathbf{w}+\sum_{i,j}\lambda_i^j\mathbf{x}_i^j-\sum_{i,j}\delta_i^j\mathbf{x}_i^{j+1}=0, \quad (4)$$

$$\frac{\partial L}{\partial a_j}=-\sum_i\lambda_i^j+1=0,\ \frac{\partial L}{\partial b_j}=\sum_i\delta_i^j-1=0, \quad (5)$$

$$\frac{\partial L}{\partial \varepsilon_i^j}=-\gamma_i^j-\lambda_i^j+C=0, \quad (6)$$

$$\frac{\partial L}{\partial \varepsilon_i^{*j+1}}=-\gamma_i^{*j+1}-\delta_i^j+C=0, \quad (7)$$

$$\begin{gathered}\lambda_i^j\geq 0,\delta_i^j\geq 0,\gamma_i^j\geq 0,\gamma_i^{*j+1}\geq 0,\\\lambda_i^j\left(\mathbf{w}^T\mathbf{x}_i^j-a_j-\varepsilon_i^j\right)=0,\\\delta_i^j\left(-\mathbf{w}^T\mathbf{x}_i^{j+1}+b_j-\varepsilon_i^{*j+1}\right)=0,\\\gamma_i^j\varepsilon_i^j=0,\gamma_i^{*j+1}\varepsilon_i^{*j+1}=0,\forall i,j.\end{gathered} \quad (8)$$

We rewrite the primal problem such that the terms that cancel each other are closer to each other,

$$L=\frac{1}{2}\|\mathbf{w}\|^2+\mathbf{w}^T\left(\sum_{i,j}\lambda_i^j\mathbf{x}_i^j-\delta_i^j\mathbf{x}_i^{j+1}\right) \tag{9}$$

$$-\sum_{i,j}(\lambda_i^j+\gamma_i^j)\varepsilon_i^j-\sum_{i,j}(\delta_i^j+\gamma_i^{*j+1})\varepsilon_i^{*j+1} \tag{10}$$

$$+C\left(\sum_{i,j}\varepsilon_i^j+\sum_{i,j}\varepsilon_i^{*j+1}\right) \tag{11}$$

$$+\sum_{j=1}^{R-1}(a_j-b_j)-\sum_{i,j}\lambda_i^j a_j+\sum_{i,j}\delta_i^j b_j. \tag{12}$$

Eq. 4 is applied to 9 and results to $-\frac{1}{2}\|\mathbf{w}\|^2$, Eqs. in 10 and 11 cancel each other when KKT conditions in Eqs. 6 and 7 are used, and the equality conditions in 5 are used to cancel the terms in 12. Conditions in Eqs. 6,7 and 8 results to, $0 \leq \lambda_i^j \leq C$ and $0 \leq \delta_i^j \leq C$. And Eqs. in 5 are used to obtain the sum to one conditions of $\sum_i \lambda_i^j=1, \sum_i \delta_i^j=1$.

To simplify notation, we rewrite $\mathbf{w}$ as $\mathbf{w} = \mathbf{Q}\mu$, where $\mathbf{Q} = [-\mathbf{X}_{1\ldots} - \mathbf{X}_{R-1}\mathbf{X}_{2\ldots}\mathbf{X}_R]$, $\mathbf{X}_j=\{\mathbf{x}_1^j,\ldots,\mathbf{x}_{n_j}^j\}$ are the samples of class j, and $\mu=[\mu_1^1,\ldots\mu_{R-1}^1\mu_1^2,\ldots,\mu_{R-1}^2]$ are all of the Lagrange multipliers, with $\mu_j^1=\{\lambda_1^j,\ldots,\lambda_{n_j}^j\}$, and $\mu_j^2=\{\delta_1^j,\ldots,\delta_{n_j}^j\}$ are the Lagrange multipliers for samples from classes j and $j+1$ penalized by hyperplanes with biases a_j and b_j, respectively.

Importantly using these equalities in (3) and the KKT conditions results in the dual problem,

$$\begin{array}{lllll} \max & W(\mu)=-\mu^T\mathbf{Q}^T\mathbf{Q}\mu & 0\leq\mu_i\leq C & , & i=1,\ldots,N_L \\ & \mathbf{1}^T\mu_j^1=1, \quad \mathbf{1}^T\mu_j^2=1 & j=1,\ldots,R-1, & & \end{array} \tag{13}$$

where $N_L = 2N - n_1 - n_r$ is the total number of Lagrange multipliers. Eq. (13) is a simple quadratic optimization problem. Unlike the algorithm proposed by [38]. This sum of margins maximizing algorithm only includes (additional) equality conditions.

The parameters—The parameter C in the above formulations control the importance of minimizing the training error versus maximizing the margin. Specifically, the amount of training error is controlled by the two conditions on the Lagrange multipliers in the dual problem, Eq. (13), i.e., the upper-bound C and the sum-to-one conditions. For instance, if $C=1/n_j^i$, where n_j^i are the number of samples in a group of Lagrange multipliers from one of $i = 1, 2$, and $j = 1, \ldots, R - 1$, then all the samples in this group will have a Lagrange multiplier equal to C to satisfy the sum-to-one condition $\mathbf{1}^T \mu_j^i=1$. This means that, all the corresponding training samples in this group will be incorrectly ordered with respect to the corresponding ranking function. On the other hand, if C is scaled to a larger value, $C=1/(\kappa n_j^i)$ with a scalar κ in $0 < \kappa < 1$, then at most κ fractions of the Lagrange multipliers will be nonzero. Hence, we use a parameter $0 < \kappa \leq 1$ to control the parameters C_j^i with respect to the number of samples in the corresponding group of multipliers, i.e., $C_j^i=\frac{1}{\kappa n_j^i}$, for each group of Lagrange multipliers. Herein, we keep a global parameter C for simplicity.

Comparison with the literature—The fixed margin strategy is defined as,

$$\begin{array}{ll} \min\limits_{\{\mathbf{w},\varepsilon_i^j,\varepsilon_i^{*j+1},b_i\}} & \frac{1}{2}\|\mathbf{w}\|^2+C\sum\limits_{j=1}^{R-1}\left(\sum\limits_{i=1}^{n_j}\varepsilon_i^j+\sum\limits_{i=1}^{n_{j+1}}\varepsilon_i^{*j+1}\right) \\ \text{such that} & \left\{\begin{array}{c} \mathbf{w}^T\mathbf{x}_i^j-b_j \leq -1+\varepsilon_i^j, \\ \mathbf{w}^T\mathbf{x}_i^{j+1}-b_j \geq 1-\varepsilon_i^{*j+1}, \\ \varepsilon_i^j \geq 0, \quad \varepsilon_i^{*j+1} \geq 0, \quad \forall i,j \end{array}\right. \end{array} \qquad (14)$$

where b_j are the biases of the hyperplanes separating the classes [38].

The main advantage of the fixed margin strategy is the simplicity of its dual problem. However, note that, in this case, the margin $1/\|\mathbf{w}\|$ is biased towards those two classes that are closest to each other. This is a typical problem in many machine learning algorithms [48], [15], [24], yielding suboptimal, biased solutions. Similarly, [6] derives an implicitly constrained fixed margin problem to improve on the accuracy of the fixed-margin strategy. Here, the errors from all samples are considered in the computation of every hyperplane, as opposed to just using the samples from neighboring classes as in the above. The limitation of the formulation is that it minimizes the error between neighboring classes, rather than the global error measure, which also results in biased solutions [15].

As an alternative to the fixed margin, the sum of margins strategy maximizes the margins between *every* neighboring class. More formally, [38] defined one possible objective as,

$$\begin{array}{ll} \min\limits_{\{\mathbf{w},\varepsilon_i^j,\varepsilon_i^{*j+1},a_i,b_i\}} & \sum\limits_{j=1}^{R-1}(a_j-b_j)+C\sum\limits_{j=1}^{R-1}\left(\sum\limits_{i=1}^{n_j}\varepsilon_i^j+\sum\limits_{i=1}^{n_{j+1}}\varepsilon_i^{*j+1}\right) \\ \text{such that} & \left\{\begin{array}{c} a_j \leq b_j, \quad b_j \leq a_{j+1}, \\ \mathbf{w}^T\mathbf{x}_i^j-a_j \leq \varepsilon_i^j, \mathbf{w}^T\mathbf{x}_i^{j+1}-b_j \geq -\varepsilon_i^{*j+1} \\ \|\mathbf{w}\|^2 \leq 1, \quad \varepsilon_i^j \geq 0, \quad \varepsilon_i^{*j+1} \geq 0, \quad \forall i,j. \end{array}\right. \end{array} \qquad (15)$$

When compared to the objective function of the fixed margin strategy in Eq. 14, the above formulation can control the margin weights separately with the $\sum_{j=1}^{R-1}(a_j - b_j)$ term. However, it is missing the $\|\mathbf{w}\|^2$ term that corresponds to the global scale of the margins.

Furthermore, the dual formulation of Eq. 15 results in a quadratic optimization problem with several inequalities. As a consequence, the solution of the problem with an iterative procedure becomes computationally expensive because of the need to keep track of a large number of inequalities at every iteration.

On the other hand, the dual problem in our formulation is simple and can be readily solved using an unsophisticated iterative approach. Furthermore, our solution considers maximizing the margin between every consecutive class and the global scale defined by $1/\|\mathbf{w}\|$. In summary, we have eliminated the drawbacks of the two previous approaches while keeping their advantages.

III. Multiple Ranking SVM (MRSVM)

We now present the extension of the above approach to multiple ranking learning and derive a computationally efficient algorithm.

A. General formulation

Learning multiple ranking functions is done by solving for each of the ranking problems simultaneously. Formally, this requires us to extend (2) as follows,

$$\min \quad F\left(\mathbf{w}_1, \varepsilon_{i_1}^{j_1}, \varepsilon_{i_1}^{*j_1+1}, a_{j_1}, b_{j_1} | C_1\right) + F\left(\mathbf{w}_2, \varepsilon_{i_2}^{j_2}, \varepsilon_{i_2}^{*j_2+1}, a_{j_2}, b_{j_2} | C_2\right).$$

Note that, an important advantage of this extension is to be able to set the relative importance of the ranking problems by adjusting the weight factors C_1 and C_2 in a single problem. This allows us to determine a solution that balances the error between the two rankings. Furthermore, it may find a more sparse solution than solving two ranking problems separately, since the support vectors can be shared between the two. In the experimental results section, we show that the simultaneous solution performs better than separate solutions would.

B. SMO solution

The Sequential Minimal Optimization (SMO) algorithm is proposed as a fast, efficient, yet simple solution to the dual problem in SVM [31]. As the kernel matrix size is too large for problems with large number of samples, it becomes impractical to solve the SVM quadratic optimization. SMO handles this by iteratively updating the solution based on two samples (selected by heuristics) and a closed-form solution at each step. After a certain number of iterations, the optimal solution can be obtained. Our SMO-based formulation results in linear and quadratic times. Additionally, the memory used by SMO scales linearly with the number of training samples.

For the ranking solution derived in the previous section, the size of the quadratic problem in (13), scales twice as fast as the number of samples. Fortunately, we show that a SMO iterative procedure does not suffer from this limitation. Our solution is detailed in the following.

We rewrite (13) by isolating two of the Lagrange multipliers for the same class *j* concerning the same hyperplane *i*. Let these multipliers be $\mu_1 = \mu^i_{j_1}$ and $\mu_2 = \mu^i_{j_2}$. Then,

$$\min_{\mu_1,\mu_2} -W = \sum_{k_1,k_2=1}^{2} \mu_{k_1}\mu_{k_2}\mathbf{K}_{k_1k_2} + 2\sum_{k_1=1}^{2}\mu_{k_1}\mathbf{z}_{k_1} + z, \qquad (16)$$

where $\mathbf{K} = \mathbf{Q}^T\mathbf{Q}$, $\mathbf{z}_{k_1} = \sum_{k_2=3}^{N_L}\mu_{k_2}\mathbf{K}_{k_1k_2}$, and $z = \sum_{k_1,k_2=3}^{N_L}\mu_{k_1}\mu_{k_2}\mathbf{K}_{k_1k_2}$. From the constraints in (13),

$$0 \le \mu_1, \mu_2 \le C, \quad \mu_1 + \mu_2 = D,$$

where $D = 1 - \sum_{k_1=3}^{n_j}\mu^i_{j_{k_1}}$. Now we insert $\mu_1 = D - \mu_2$ to Eq. (16), and set the derivative of W with respect to μ_2 to zero, we have $\frac{\partial W}{\partial \mu_2} = -2(D-\mu_2)\mathbf{K}_{11} + 2(D-2\mu_2)\mathbf{K}_{12} + 2\mu_2\mathbf{K}_{22} - 2\mathbf{z}_1 + 2\mathbf{z}_2 = 0$. From this we can obtain the updated Lagrange multiplier as,

$$\mu_2^{new} = \frac{D(\mathbf{K}_{11} - \mathbf{K}_{12}) + \mathbf{z}_1 - \mathbf{z}_2}{\mathbf{K}_{11} + \mathbf{K}_{22} - 2\mathbf{K}_{12}}.$$

For clarity, the above equation can be rewritten as

$$\mu_2^{new} = \mu_2 + \frac{u_1 - u_2}{\mathbf{K}_{11} + \mathbf{K}_{22} - 2\mathbf{K}_{12}}, \qquad (17)$$

where $u_k = \mathbf{K}_{1k}\mu_1 + \mathbf{K}_{2k}\mu_2 + \mathbf{z}_k$ is the projection of the sample to the weighted direction **w**, i.e., $u_k = \mathbf{w}^T\mathbf{x}_k^{j_2}$ for $i = 1$, and $u_k = -\mathbf{w}^T\mathbf{x}_k^{j_2+1}$ for $i = 2$, where $i = \{1, 2\}$ indicates the Lagrange multipliers for samples that are penalized by the corresponding hyperplane. These indices are previously reflected in Eq. 13 and its derivations as the superscripts of μ^i_j.

Next, we show that the updated multiplier satisfies the boundary conditions defined in (13). These conditions are given by the minimum and maximum possible solutions defined by the equality condition $\sum_{k=1}^{n_j}\mu^i_{j_k} = 1$, i.e. $\mu_1 + \mu_2 = D$, within the feasible region $0 \le \mu_1, \mu_2 \le C$. These are the lower- and upper-bounds, $L = max(0, D - C)$ and $H = min(C, D)$, respectively.

Following these derivations, we update μ_2^{new} to the closest point in the solution space. That is,

$$\mu_2^{new}=\begin{cases} L & \text{if} \quad \mu_2^{new}<L \\ \mu_2^{new} & \text{if} \quad L \le \mu_2^{new} \le H \\ H & \text{if} \quad \mu_2^{new}>H. \end{cases} \qquad (18)$$

μ_1^{new} is given by the equality condition, $\mu_1^{new}=D-\mu_2^{new}$. After each iteration, the bias of the hyperplanes are updated if the multipliers satisfy $0<\mu_k^{new}<C$, i.e., $a_j=\mathbf{w}^T\mathbf{x}_k^j$, for = 1, and $b_j=\mathbf{w}^T\mathbf{x}_k^{j+1}$, for $i = 2$.

The main difference of the above derived SMO algorithm and the original SMO specifically designed for the two-class SVM classifier [31] or the novelty detection [35] or for estimating the support of a high-dimensional distribution [34] is the selection of the two Lagrange multipliers. Because of the sum-to-one condition for the Lagrange multipliers, the two multipliers should be selected from the group of samples from the same class rank and is penalized by the same hyperplane, i.e., the groups defined by μ_j^i. This group-based multiplier selection property allows us to readily extend the SMO for single ranking SVM to that of multiple ordering SVM by simply considering a group from one of the ordering problems at each iteration. The group and the Lagrange multipliers to be optimized at each iteration are selected using the same heuristics as in SMO for SVM.

The derived approach is summarized in Algorithm 1.

Algorithm 1

SMO solution for MRSVM

```
do
    Find a variable $\mu_2$ that violates the KKT optimality conditions.
    Search for the second variable $\mu_1$ within the same ranking problem ($d = \{1, 2\}$) and within the same group $\mu_j^i$ of $\mu_2$.
    Select the second variable that maximize the step size ($u_1 - u_2$ in (17)) and calculate the local minima for $\mu_2^{new}$ using (17).
    Update $\mu_2^{new}$ to the closest point in the solution space using (18).
    Calculate $\mu_1^{new}=D-\mu_2^{new}$
    If $0<\mu_j^{new}<C_d$, update the corresponding hyperplane biases $a_j=\mathbf{w}^T\mathbf{x}_k^j$ for $i = 1$ and $b_j=\mathbf{w}^T\mathbf{x}_k^{j+1}$ for $i = 2$.
until all variables satisfy KKT optimality conditions
```

IV. Experiments

In this section, we provide experiments illustrating the efficiency and accuracy of the proposed algorithm. We compare the accuracy of the orderings given by the derived Multiple Ranking SVM (MRSVM) and Multiple Single Ranking SVM (MSRSVM) and compared them to those given by Support Vector Regression (SVR) [39], [42] and ranking SVM (SVM$_{rank}$) [19] algorithms using standard UCI datasets. SVM$_{rank}$ provides state-of-the-art optimized implementation of fixed margin strategy based solutions.

Two applications of interest of the derived algorithm are in attribute-based object recognition and data visualization. These are given at the end of this section.

A. Computational Comparison

We illustrate the computational efficiency of Algorithm 1 using the example ranking problem shown in Fig. 2. The 3-class ranking problem is formulated as the quadratic optimization problem in (13) and the solutions are obtained by our MATLAB implementation of Algorithm 1 and the standard MATLAB implementation of the Quadratic Programming (QP) solver, both in a PC with 2.2GHz CPU. The QP solver is based on the active-set strategy [14]. Samples from 3 classes are obtained from 3 Gaussian distributions with means at $(-10, 0)$ $(5, 0)$ and $(30, 0)$ with covariance matrices equal to $2\mathbf{I}$. We run 10-fold cross-validation experiments with $\{10, 20, \dots, 100\}$ samples per class, which results in $\{27, 54, \dots, 270\}$ training samples for each validation. The same solutions are obtained with both algorithms in all of the cross-validation tests, while showing different computational times as illustrated in Fig. 3. While the time complexity of the standard QP solver scales polynomially with the number of samples, Algorithm 1 does not show a significant change. This is because, Algorithm 1 is an iterative SMO-type approach where its convergence significantly depends on the complexity of the learning problem. If the ranking classes are well separated from each other the convergence may take only a few iterations. A problem that requires a highly nonlinear solution may have a computational complexity as much as a quadratic problem. The time complexity of our solution scales between linear and quadratic with respect to the dataset size. Complexity of the ranking problem defines the average time complexity. Since the difficulty of the ranking problem does not increase by adding samples to the training set, the convergence rate and computational expense of our solution stays the same.

B. Fixed Margin Strategy vs. MSRSVM

The fixed margin strategy objective formulated in Eq. 14 mainly focuses on maximizing the margin between the closest class pairs. This results in solutions that are driven by the closest classes, which means other classes do not contribute much in computing the solution. MSRSVM algorithm solves Eq. 2 and maximizes the sum of margins. It properly assigns the same importance to maximizing the margin between each class pairs.

We illustrate these two solutions to ordinal regression in a 3-class ranking experiment. For simplicity, we designed the experiment for linearly separable ranking problems. 50 instances are sampled from each of 3 Gaussian distributions with means at $(-15, 0)$, $(0, 0)$ and

$r(\mathrm{COS}(\theta), sin(\theta))$ where $r = \{15, 20, 25, 30\}$ is the radius and $\theta = \{0, \pi/8, \pi/4\}$ is the angular degrees. Fig. 4 illustrates the configuration of these three-class ranking problem.

We solve these ranking problems using Eq. 14 and the derived MSRSVM algorithm, and calculate the total margin obtained by the solutions. This is done for ten random sample sets for each configuration of r and θ and average margins are calculated. With a fixed margin, the margin size is given by $4/\|\mathbf{w}\|$, while the sum of margins is calculated with $\Sigma_i(b_i - a_i)/\|\mathbf{w}\|$. Note that, the size of the margin controls the upper bound of the VC dimension [46]. The larger the margin the smaller the upper bound, which means the solution has smaller generalization error, i.e. the solution with larger margin is expected to yield a smaller classification error when testing previously unseen feature vectors.

Table I compares the averaged margins that are obtained with the fixed margin approach and that derived in the present paper. As seen in this figure, for all the tested cases, the margin size obtained by fixed-margin strategy is around 10, which is approximately the margin between the closest class pairs. When class 3 moves away from class 2 the margins obtained by our solution is larger than the fixed margin strategy. This is pretty important when class 3 is rotated to a vertical location, where the 1-dimensional ranking function that has the largest margin is not aligned with the horizontal line. This was illustrated in Fig. 2, where the fixed margin is still tuned to the closest class pairs and obtains a solution that is biased by this class pair and neglects the separability between classes 2 and 3. On the other hand, MSRSVM considers the margin between each consecutive rank pair and obtains a solution that has significantly larger margins than the previously discussed solutions.

C. UCI datasets

We provide a comparison with three datasets typically employed in regression problems. These are from the UCI repository[2]. For each dataset, multiple rankings are provided for two output variables. These ordinal variables are obtained by binning the continuous variables to equal frequencies, i.e., the number of samples for each class are the same. Specifically, in California housing *medianIncome* and *medianHouseValue* features, in Boston *rm* features that stands for the average number of rooms per dwelling and housing values of *class* features, and for auto-mpg *cylinders* and the mpg values in *class* features are converted to ordinal variables. In our experiments, we have used the radial basis kernel function. The parameters of the optimization problem C_1 and C_2 are selected from the set of {0.1, 1, 10, 100} using 5-fold cross-validation within the training set, both for the ordinal learning and regression problems.

Table II summarizes the dimensionality of the datasets, number of samples in training/testing, and average absolute error for each of the algorithms. The average error rates are obtained from 10 random partitions of the dataset into training and testing.

In addition, we compared our results to SVM_{rank} [19], which has been commonly used in information retrieval. As we summarized in the introduction, this ranking algorithm considers the pairwise ordered samples. Hence, to run this algorithm, we convert our

---

[2] These datasets are available at http://www.liaad.up.pt/l~torgo/Regression/DataSets.html

samples and the associated ordinal variables to a proper representation. A training sample with a ranking of r_i is represented with respect to its ranking relations to its 10 nearest neighbors, i.e., $\mathbf{x}_{ij}$ with rankings of r_{ij} for $j = \{1,\dots, 10\}$. This results in 10 pairwise preference constraints with relevance scores between the pairwise samples defined by the absolute values of the difference between the ranking labels $|r_i - r_{ij}|$. The smaller this score is, the more similar the rankings of the pairwise samples are. A ranking function is learned to accurately sort these samples and their relation to each other. The parameters of the algorithm are selected with a 5-fold cross-validation test. The ranking of a testing sample is obtained from the rank of the training sample that has the closest ranking score to the testing sample. The mean absolute errors obtained with this approach are shown in Table II. SVM_{rank} performs worse than the proposed algorithms, since it is mostly concerned with correctly ranking individual samples, while our ranking algorithms attempt to maximize the correct ordering of classes.

D. Detecting Unseen Object Classes

A key feature that significantly differentiates the human visual system from machine vision is the generalization capability to unseen object classes. Towards this goal, Lampert et.al. [21] recently proposed attribute-based object classification. Objects are described with semantic attributes (i.e. high-level object descriptions) that are common across classes, such as physical properties of animals. Lampert et. al. propose to use binary SVM classifiers to estimate the existence of an attribute in a query object. However, the binary representation of class-attribute associations is usually insufficient, since one needs to describe an attribute with more than two levels, e.g., to consider anchovy, tuna and whale, we need three levels (small, medium, and large) to discriminate them. In addition, these high level descriptions are defined by human preferences, which require ordinal classification algorithms. This is because, the metric underlying human preferences is not known, while ranking of the attribute class associations can be accurately measured. [28] applied a formulation of the ranking SVM [18] to estimate the relative orders of the attributes. However, as summarized before, ranking SVM is not derived to *estimate the ordinal variables*. This was illustrated by the SVM_{rank} algorithm in the previous section.

Direct Attribute Prediction—As a baseline, we compare our algorithm with the Direct Attribute Prediction (DAP) algorithm proposed in [21]. DAP assumes that a set of binary high-level attributes are provided for a set of training and testing classes, i.e., binary class-attribute associations are known. The relation between the high-level attributes and low-level image features are learned using the training classes and samples. A test sample is classified to one of the unseen test classes based on the attribute predictions and known binary class-attribute associations. Specifically, the posterior probability of observing an attribute a_m^c of the unseen test class c given a sample $\mathbf{x}$ is obtained by $p(a_m^c|x)=1/exp(A_m g_m(\mathbf{x})+B_m)$, where $g_m(\mathbf{x})$ is a binary SVM classifier and the Platt scaling [32] coefficients A_m and B_m are estimated with a validation set to convert the binary classifier outputs to posterior probabilities with a sigmoid function. The class label of the unseen sample is estimated using these posterior probabilities of the attribute class associations, i.e., the sample $\mathbf{x}$ is classified to the class with maximum product of MAP predictions. Formally,

$\arg\max_{c=1,\ldots,C}\prod_{m=1}^{M}p(a_m^c|\mathbf{x})$ is the decision rule that maps the input samples to one of the *C unseen test classes*. Note that, in this formulation, a_m^c are the binary variables that are obtained by thresholding the attribute-class association matrix.

Direct Attribute Prediction with Multiple Ordinal Regression—We illustrated this application of the herein derived algorithm at the beginning of this paper, Fig. 1. We use the human orderings and the proposed MRSVM algorithm to learn a ranking function $f_m(.)$ for each of m attributes. These ranking functions thus determine the ordinal classification. The estimated multiple rankings of a test sample **x** are used to assign the query to one of C classes using a probabilistic decision rule. To do this, we first estimate a Normal distribution $\mathcal{N}(\mu_{r_m}, \sigma_{r_m})$ from the ranking estimations $f_m(\mathbf{x}_V|r_m)$ of the same validation set samples $\mathbf{x}_V$ as used for DAP (i.e., the validation set samples that are used to estimate the Platt scaling above) and given by the ranking labels of r_m. The likelihood of a new sample **x** of rank r_m for attribute m is simply given by,

$$p(\mathbf{x}|r_m)=\frac{1}{\sqrt{2\pi\sigma_{r_m}^2}}exp\left(-\frac{\|f_m(\mathbf{x}|r_m)-\mu_{r_m}\|^2}{2\sigma_{r_m}^2}\right). \quad (19)$$

The MAP prediction of $p(r_m^c|\mathbf{x})$ is obtained by the Bayes' rule,

$$p(r_m^c|\mathbf{x})=\frac{p(\mathbf{x}|r_m^c)p(r_m^c)}{\sum_{r_m=1}^{R_m}p(\mathbf{x}|r_m)p(r_m)},$$

where r_m^c is the ordinal attribute value r_m of class c, the attribute rank priors $p(r_m)$ are assumed to be equal, and the likelihood probability $p(\mathbf{x}|r_m)$ of observing sample **x** given attribute m and ranking r_m is obtained from (19).

A test sample **x** is classified to the class that has the maximum sum-of-posterior-probabilities,

$$\arg\max_{c=1,\ldots,C}\sum_{m=1}^{M}p(r_m^c|\mathbf{x}).$$

We call this approach Direct Attribute Prediction (DAP) with MSRSVM. Note that, the main advantage of our approach (compared to the baseline algorithm derived above) is its ability to learn multiple ranking functions $f_m(.)$ instead of the binary classifiers $g_m(.)$ employed by other approaches.

Dataset and Experimental Setup—We used the "Animals with Attributes" dataset [21] to test the performance of the algorithms detailed above. The dataset includes 50 animal classes with a partition of 40 training and 10 testing classes. In our experiments, we used 92 samples/class for a total of 3, 680 training and 920 testing samples. Each class is associated with each of 85 attributes with a continuous scalar variable collected from human subjects. Images of the objects are sampled from the Internet and are represented as a set of 6 features of RGB color histograms, SIFT, rgSIFT, PHOG, SURF and local self-similarity histograms.

DAP and DAP with MSRSVM are used to learn binary and multiple ranking functions between the 40 training classes and their corresponding attribute associations. The continuous attribute-class association scores are converted to binary variables via thresholding with the average value [21]. The multiple rankings are obtained by sorting the class association scores for each attribute and dividing into $R = 4$ ranking levels such that the R regions have the same area under the curve of sorted association scores. This conversion allows similar association scores to share the same ranking value. We use exponential-χ^2 kernels [40] and calculate the average of them to combine the kernel values calculated using 6 features of histograms. Formally, we use the kernel $k(\mathbf{x},\mathbf{y})=\frac{1}{6}\sum_{i=1}^{6} exp(-\chi^2(\mathbf{x}_i,\mathbf{y}_i)/\gamma_i)$ that averages exponential-χ^2 kernels of $exp(-\chi^2(\mathbf{x}_i, \mathbf{y}_i)/\gamma_i)$ with the kernel scale parameter of γ_i that are based on the χ^2 distances between the i^{th} feature vectors $\mathbf{x}_i$ and $\mathbf{y}_i$, where $\chi^2(\mathbf{x}_i,\mathbf{y}_i)=\sum_{j=1}^{d}(\mathbf{x}_{ij}-\mathbf{y}_{\mathbf{ij}})^2/(\mathbf{x}_{ij}+\mathbf{y}_{\mathbf{ij}})$ and d is the dimensionality of the i^{th} feature vectors.

The parameters of the algorithms are selected by a 10-fold cross-validation experiment in the training set. This resulted in the selection of the exponential-χ^2 kernel widths γ_i to be the median of all pairwise χ^2 distances out of the set of parameters of scaled medians $\{5^{-1}, 5^{-1/2}, \ldots, 5^1\} * median$. The parameter C as defined in the SVM formulation of LIBSVM [2] is selected to be 1 from the set of {0.1, 1, 10, 100}, and the parameter κ in MSRSVM is set to .001.

Results—Table III shows the recognition rates of unseen testing samples obtained with DAP and DAP with MSRSVM. We provided the experimental results for 10 independent validation sets, and their mean. We also show the average confusion tables over 10 runs on independent validation sets in Fig. 5. As seen in the figure, our proposed ranking solution provides less confusions than the binary classification approach. We also computed the average ratio of confused samples to correctly classified samples by $\frac{1}{10}\sum_{i=1}^{10}\frac{1-c_{ii}}{c_{ii}}$, where c_{ii} is the i^{th} diagonal element of the confusion table. This ratio is 25.81 for DAP and 2.83 for DAP with MSRSVM.

Ordinal Attribute Regression—We have also experimented with several alternative ways of estimating the ordinal attribute values, i.e., learning various ranking functions $f_m(.)$ from the low-level image features. We use ordinal attribute variables to represent the class-attribute associations. These ordinal variables are obtained by binning the continuous variables to equal frequencies. We have tested the accuracy of estimating these variables using three algorithms. Although Multiclass SVM (*M-SVM*) minimizes the misclassification rate rather than misranking, we provide comparisons to this algorithm to clarify the differences between the classification and ranking algorithms. *Gaussian Processes* *for Ordinal Regression* (GPOR) [6] is formulated using a Bayesian framework, which allows to use a gradient descent based optimization algorithm to estimate the model parameters such as ranking hyperplane biases, kernel parameter, and noise level. This parameter estimation may be useful in several problems. However, as pointed out by [6], the scalability of this algorithm is limited due to gradient descent algorithm that becomes the computational bottleneck in large scale problems. We compared the performance of these two algorithms to the proposed MSRSVM solution in estimating the ordinal attribute values. The parameters for M-SVM and MSRSVM are obtained with 10-fold cross validation on the training set, while GPOR utilizes a gradient descent based optimization to estimate its parameters. As seen in Table IV, MSRSVM obtained lower mean absolute errors than M-SVM and GPOR with smaller standard deviation over all attribute label estimations. M-SVM performed worst as its objective is not defined for ranking problems. The limitation of the GPOR performance is most likely due to approximations in model based solution. GPOR training time is much larger than SMO based solutions of M-SVM and MSRSVM. This is because the gradient descent technique with variable convergence rates depends on the complexity of the ranking problem.

E. Visualizing Multiple Rankings of Attributes

Another application of MRSVM algorithm is to order objects based on *multiple* visual properties. This can be used, for example, to give visual feedback to a user’s search.

To illustrate this application, we have performed the following experiment using the “Animals with Attributes” data-set [21]. We selected the *whiteness* and *brownness* attribute class associations and converted these measurements to ordinal variables defined by 4 underlying ranking functions. Our goal is to learn the multiple ranking function in the training set classes such that it accurately orders the unseen testing class objects based on estimation of the selected class attribute. We used the training and testing sets that were described in the previous section, and use the 2, 688-dimensional multi-resolution RGB histograms to represent the images.

The mean absolute errors are shown in Table V, where we see a significant error rate for the derived approach. The MRSVM is then applied to unseen test samples to estimate the attribute rankings and visualize them with respect to the ranking scores. The samples from the classes with maximum relevance scores are also shown in Table VI. Brownness and whiteness of the animals increase from top to bottom and left to right, respectively. The number of animal images associated with a certain rank are shown in parentheses. The selected images of the 3 sample animals are in Fig. 6. This provides a visual interpretation of the order of the objects with respect to the attributes. We see that the animals that have darker skins (e.g. leopard, hippopotamus and chimpanzee) ranked 1 in whiteness (i.e., least white), as expected.

Recent studies in action recognition [5] are very similar to the set up defined above and are thus also poised to benefit from the algorithms derived in the present paper.

V. Conclusion

The goal of the algorithms derived in the present paper is to estimate a function that orders the objects with respect to a set of specified preferences. To the authors knowledge, this paper presents the first algorithm that can learn multiple ranking functions minimizing the misclassification risk between *all* neighboring classes, yielding superior results to state of the art algorithms. In particular, we have derived a SMO-based iterative solution that allows learning of the functions from very large databases. Applications in several data-sets shows that the proposed ranking algorithm performs better than Support Vector Regression and SVM_{rank}. An important application of the derived algorithm is to estimate the attribute associations of an object based on multiple features. This provides a rapid and efficient solution to one of the most classical (yet challenging) problems in machine learning and computer vision – object categorization.

Acknowledgments

This research was supported in part by the National Institutes of Health, R01-EY-020834, and a Google Faculty Award.

References

1. Cao, Y.; Xu, J.; Liu, T.; Li, H.; Huang, Y.; Hon, H. Adapting ranking SVM to document retrieval. Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; ACM; 2006. p. 186-193.
2. Chang C, Lin C. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2011; 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
3. Chapelle, O.; Chang, Y. Yahoo! learning to rank challenge overview. JMLR: Workshop and Conference Proceedings; 2011. p. 1-24.
4. Chapelle O, Keerthi S. Efficient algorithms for ranking with svms. Information retrieval. 2010; 13(3):201–215.
5. Chen, W.; Xiong, C.; Xu, R.; Corso, JJ. Actionness ranking with lattice conditional ordinal random fields. Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on; IEEE; 2014. p. 748-755.
6. Chu, W.; Keerthi, S. New approaches to support vector ordinal regression. Proceedings of the 22nd International Conference on Machine Learning; ACM; 2005. p. 145-152.
7. Clémençon S, Depecker M, Vayatis N. Ranking forests. The Journal of Machine Learning Research. 2013; 14(1):39–73.
8. Cohn JF, De la Torre F. Automated face analysis for affective. The Oxford Handbook of Affective Computing. 2014; 131
9. Du S, Tao Y, Martinez AM. Compound facial expressions of emotion. Proceedings of the National Academy of Sciences. 2014; 111(15):E1454–E1462.
10. Duchi JC, Mackey L, Jordan MI, et al. The asymptotics of ranking algorithms. The Annals of Statistics. 2013; 41(5):2292–2323.
11. Fernandez-Navarro F, Gutierrez P, Hervas-Martinez C, Yao X. Negative correlation ensemble learning for ordinal regression. Neural Networks and Learning Systems, IEEE Transactions on. 2013; 24(11):1836–1849.
12. Fernandez-Navarro F, Riccardi A, Carloni S. Ordinal neural networks without iterative tuning. Neural Networks and Learning Systems, IEEE Transactions on. 2014; 25(11):2075–2085.
13. Frank, E.; Hall, M. A simple approach to ordinal classification. Springer; 2001.

14. Gill, P.; Murray, W.; Wright, M. Numerical Linear Algebra and Optimization. Vol. 1. Addison Wesley; 1991.
15. Hamsici OC, Martinez AM. Bayes optimality in linear discriminant analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2008; 30(4):647–657.
16. Hastie, T.; Tibshirani, R.; Friedman, J.; Franklin, J. The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer; 2005.
17. Herbrich, R.; Graepel, T.; Obermayer, K. Advances in Neural Information Processing Systems. MIT; 1999. Large margin rank boundaries for ordinal regression; p. 115-132.
18. Joachims, T. Optimizing search engines using clickthrough data. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; ACM; 2002. p. 133-142.
19. Joachims, T. Training linear svms in linear time. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining; ACM; 2006. p. 217-226.
20. Kramer S, Widmer G, Pfahringer B, De Groeve M. Prediction of ordinal classes using regression trees. Fundamenta Informaticae. 2001; 47(1):1–13.
21. Lampert, C.; Nickisch, H.; Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR); Miami, FL. 2009.
22. Liu, T.; Xu, J.; Qin, T.; Xiong, W.; Li, H. Letor: Benchmark dataset for research on learning to rank for information retrieval. Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval; 2007. p. 3-10.
23. Martinez AM, Du S. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. The Journal of Machine Learning Research. 2012; 98888:1589–1608. [PubMed: 23950695]
24. Martinez AM, Zhu M. Where are linear feature extraction methods applicable? Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2005; 27(12):1934–1944.
25. McCullagh P. Regression models for ordinal data. Journal of the Royal Statistical Society, Series B. 1980; 42(2):109–142.
26. Myers, A.; Teo, CL.; Fermüller, C.; Aloimonos, Y. Affordance detection of tool parts from geometric features. IEEE International Conference on Robotics and Automation (ICRA); 2015.
27. Neth D, Martinez AM. Emotion perception in emotionless face images suggests a norm-based representation. Journal of Vision. 2009; 9(2):1–11. [PubMed: 19271875]
28. Parikh, D.; Grauman, K. Relative attributes. Proceedings of IEEE International Conference on Computer Vision (ICCV); IEEE; 2011.
29. Perez-Ortiz M, Gutierrez P, Hervas-Martinez C. Projection-based ensemble learning for ordinal regression. Cybernetics, IEEE Transactions on. May; 2014 44(5):681–694.
30. Perez-Ortiz M, Gutierrez P, Hervas-Martinez C, Yao X. Graph-based approaches for over-sampling in the context of ordinal regression. Knowledge and Data Engineering, IEEE Transactions on. May; 2015 27(5):1233–1245.
31. Platt, J. Advances in Kernel Methods. MIT press; 1999. Fast training of support vector machines using sequential minimal optimization; p. 185-208.
32. Platt, J. Advances in Large Margin Classifiers. MIT press; 2000. Probabilities for SV machines.
33. Rajkumar, A.; Agarwal, S. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. Proceedings of the 31st International Conference on Machine Learning; 2014.
34. Schölkopf B, Platt J, Shawe-Taylor J, Smola A, Williamson R. Estimating the support of a high-dimensional distribution. Neural Computation. 2001; 13(7):1443–1471. [PubMed: 11440593]
35. Schölkopf B, Williamson R, Smola A, Shawe-Taylor J, Platt J. Support vector method for novelty detection. Advances in Neural Information Processing Systems. 2000; 12(3):582–588.
36. Sculley, D. Combined regression and ranking. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining; ACM; 2010. p. 979-988.
37. Seah C-W, Tsang I, Ong Y-S. Transductive ordinal regression. Neural Networks and Learning Systems, IEEE Transactions on. Jul; 2012 23(7):1074–1086.

38. Shashua A, Levin A. Ranking with large margin principle: Two approaches. Advances in Neural Information Processing Systems. 2003:961–968.
39. Smola A, Schölkopf B. A Tutorial on Support Vector Regression. Statistics and computing. 2004; 14(3):199–222.
40. Sreekanth, V.; Vedaldi, A.; Zisserman, A.; Jawahar, C. Generalized rbf feature maps for efficient detection. Proceedings of British Machine Vision Conference; 2010.
41. The Angiosperm Phylogeny Group. An ordinal classification for the families of flowering plants. Annals of the Missouri Botanical Garden. 1998:531–553.
42. Tsai, Y-H.; Hamsici, OC.; Yang, M-H. Adaptive region pooling for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 731-739.
43. Turney, PD. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics; 2002. p. 417-424.
44. Uematsu, K.; Lee, Y. Technical Report. Vol. 873. Department of Statistics, The Ohio State University; 2013. Statistical optimality in multipartite ranking and ordinal regression.
45. Ullman S, Vidal-Naquet M, Sali E. Visual features of intermediate complexity and their use in classification. Nature Neuroscience. 2002; 5(7):682–687. [PubMed: 12055634]
46. Vapnik, V. Statistical learning theory. Wiley; New York: 1998.
47. You D, Benitez-Quiroz CF, Martinez AM. Multiobjective optimization for model selection in kernel methods in regression. Neural Networks and Learning Systems, IEEE Transactions on. 2014; 25(10):1879–1893.
48. You D, Hamsici OC, Martinez AM. Kernel optimization in discriminant analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2011; 33(3):631–638.
49. Zheng Z, Zha H, Zhang T, Chapelle O, Chen K, Sun G. A general boosting method and its application to learning ranking functions for web search. Advances in Neural Information Processing Systems. 2008; 20:1697–1704.

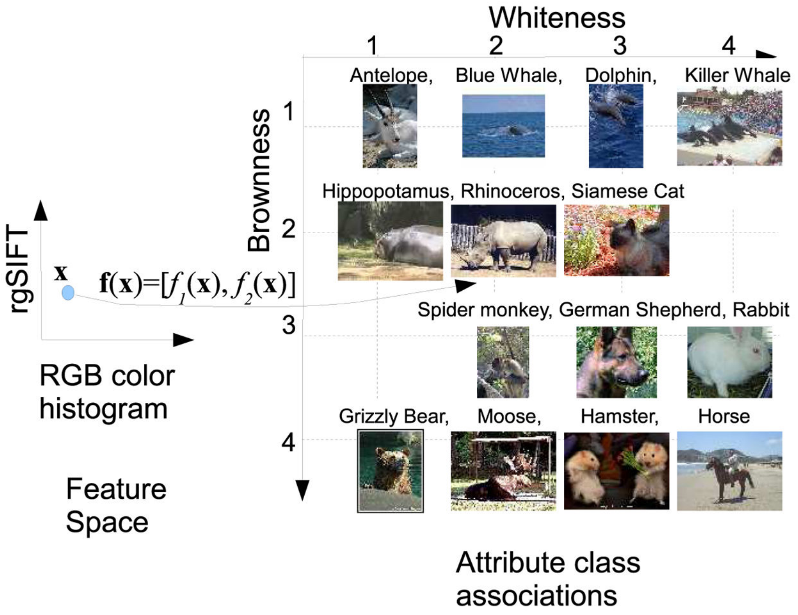
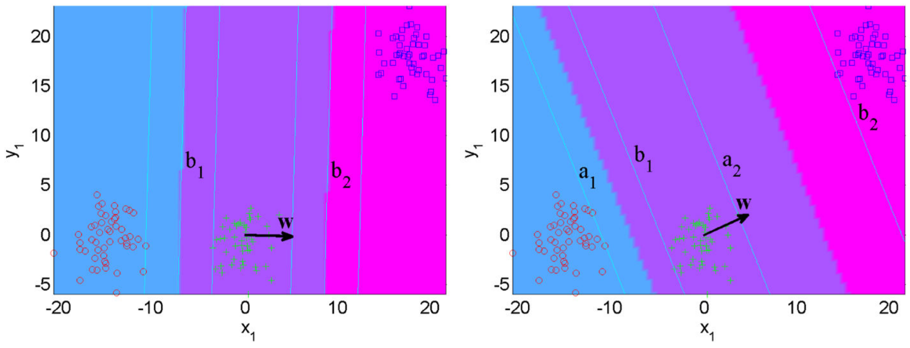


Fig. 1.

Here we illustrate the use of multiple ranking functions in the classification of an unseen sample **x**. Attribute-class associations are predefined according to human subjects' attribute preferences (whiteness and brownness on a scale from 1 to 4) for a set of animal classes. Our multiple ranking SVM is used to learn the function **f**(.) from a training set to estimate the multiple attribute rankings from samples. The attribute associations of the test sample **x** are estimated with the multiple ranking functions, i.e. $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. These rankings are used to assign the sample to one of the unseen testing classes.



(a)

(b)

Fig. 2.

Different learning strategies result in distinct rankings. (a) Shows the fixed margin strategy, where the margin is defined by the closest classes. In this case the goal is to find the two hyperplanes with the common direction vector **w** and biases b_1 and b_2 that maximize the minimum margin. Our sum of margins strategy shown in (b) extracts the direction **w** that maximizes the sums of margins $\sum_{i=1}^{2}(b_i - a_i)$ with the four hyperplane biases of a_1, b_1, a_2, and b_2. Different colors correspond to classification regions defined by the ranking rule. The solution of fixed margin strategy is biased by the closest class pairs of 1 and 2, and the sum of margin strategy obtains a larger margin by maximizing the sum of margins.

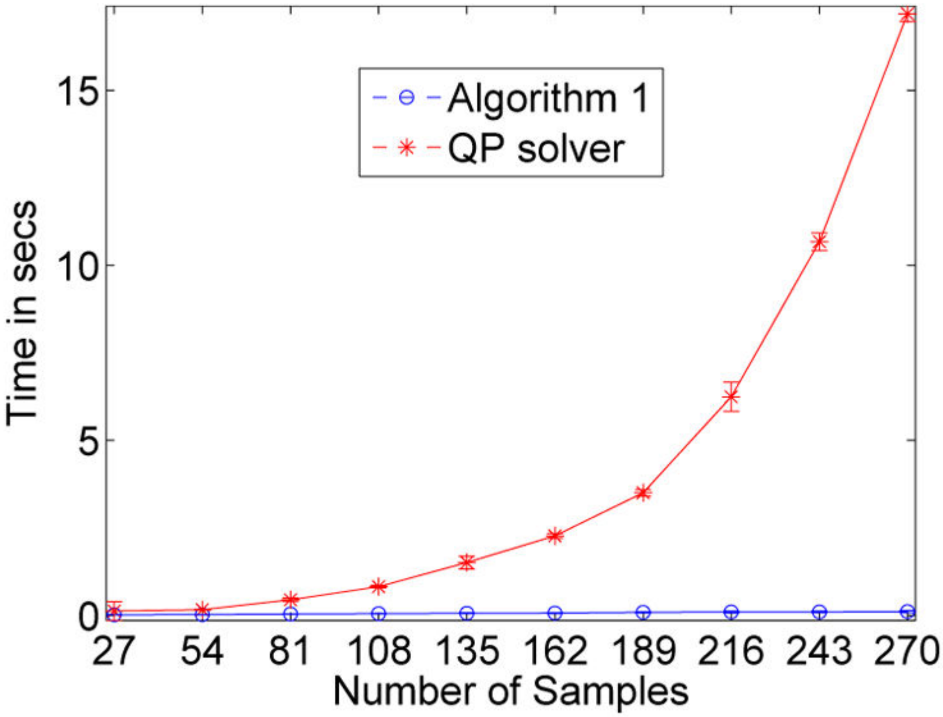


Fig. 3.

Computational times required to solve the quadratic optimization problem in (13) for the 3-class ranking problem illustrated in Fig. 2. Algorithm 1 is compared to a standard MATLAB Quadratic Programming (QP) solver with {27, 54,…, 270} training samples in each of the 10-fold cross validation experiments. Average and standard deviations of the computational times are plotted.

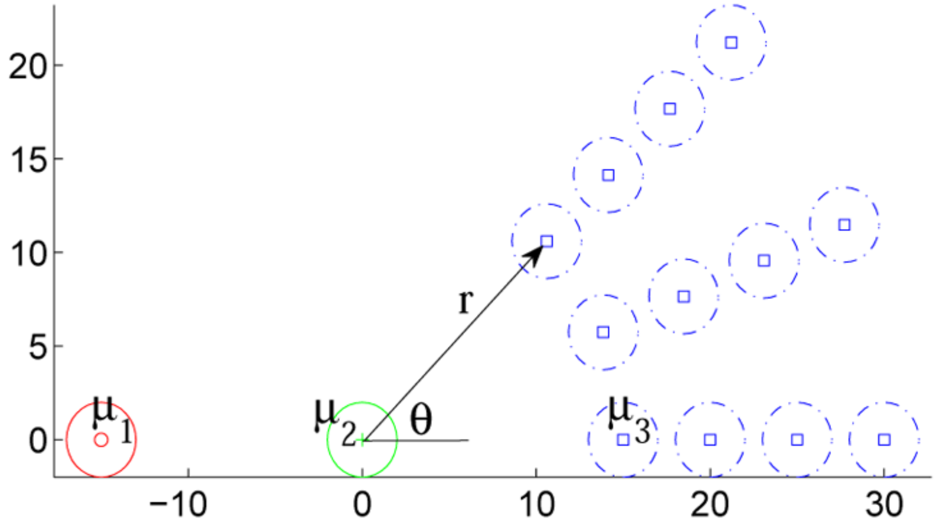


Fig. 4.

Here we illustrate the set of ranking problems that are used to compare the fixed margin strategy with MSRSVM. Classes are sampled from 3 Gaussian distributions of $N(\mu_{\mathbf{i}}, 4\mathbf{I})$, where $\mu_1 = (-15, 0)^T$, $\mu_2 = (0, 0)^T$, and the covariance matrices with scaled identity matrices of $4\mathbf{I}$. Class 3 is simulated to be at multiple locations of $\mu_3 = r(cos(\theta), sin(\theta))^T$, for $r = \{15, 20, 25, 30\}$ and $\theta = \{0, \pi/8, \pi/4\}$. The circles correspond to Gaussian distributions, with $4\mathbf{I}$ as covariance matrices. Classes 1 and 2 are fixed and represented with circles of solid lines. Class 3 is set to each of the locations (r, θ), which are represented with dashed circles.

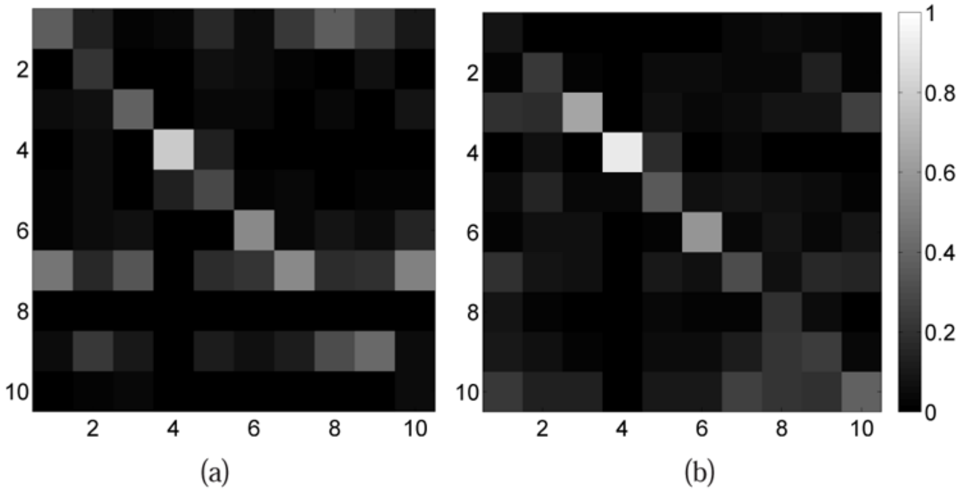


Fig. 5.

Shown here are the average *confusion tables* for the (a) DAP and (b) DAP with MSRSVM approaches. These results are obtained by averaging 10 runs over independent validation sets. The indices of the rows and columns correspond to the indices of the testing classes.

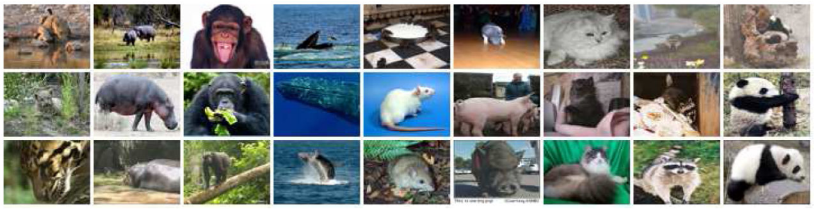


Fig. 6.

Three sample images of the animals in each nonempty row and column of the multiple ranking table. Rows from left to right corresponds to column wise scan of the nonempty cells of Table VI, i.e. leopard, hippopotamus, chimpanzee, humpback whale, rat, pig, persian cat, raccoon, giant panda.

TABLE I

A comparison between the margins obtained using different approaches. The values in the parenthesis (a,b) mean: fixed margin strategy in Eq. 14 for (a) and MSRSVM algorithm for (b). A set of ranking problems are tested by varying the location of class 3 according to r and θ as shown in Fig. 4.

θ/r	15	20	25	30
0	(10.06, 11.39)	(12.88, 17.91)	(12.52, 18.28)	(11.29, 26.96)
$\pi/8$	(10.24, 12.45)	(14.44, 18.92)	(11.29, 21.39)	(12.86, 25.79)
$\pi/4$	(9.17, 9.72)	(11.36, 12.03)	(12.24, 26.5)	(11.72, 25.21)

TABLE II

Shown here are the mean absolute errors $\frac{1}{N}\sum_{i=1}^{N}\|r_i - \hat{r}_i\|$ between the ground truth ranking of r_i and the estimated ranking of $\hat{r}_i$ using MRSVM, MSRSVM, SVR, and ranking SVM based on pairwise orders (SVM$_{rank}$). We also provide the training and testing partitions, the dimensionality (*d*) of the dataset, the labels of the database features, the number of ranking classes (R) and the ranking results (mean absolute errors) for individual dimensions and their average.

Dataset	**train/test (d)**	**feature (R)**	**Methods**			
			MRSVM	**MSRSVM**	**SVR**	**SVM$_{rank}$**
California Housing	1000/19640 (7)	medianIncome (4)	0.81	0.76	1.2	1.15
		medianHouseValue (4)	0.83	0.83	1.03	1.15
		avg	0.82	0.79	1.12	1.15
Boston	300/206 (12)	rm (4)	0.84	0.84	1.14	1.21,
		class (4)	0.48	0.51	0.88	1.05
		avg	0.66	0.68	1.01	1.13
Auto-mpg	192/200 (6)	cylinders (3)	0.08	0.09	0.25	0.33
		class (5)	0.41	0.50	0.68	1.19
		avg	0.24	0.29	0.46	0.76

TABLE III

Here we compare the recognition rates of unseen testing set samples obtained by DAP and DAP with MSRSVM algorithms which are trained with 10 independent validation sets of training set.

Method/Validation Set	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**mean**
DAP	35.87	36.2	36.63	35.65	35.65	36.63	36.3	37.72	36.09	35.43	36.21
DAP w MSRSVM	38.48	39.13	38.26	40.76	40.65	39.67	39.57	38.15	39.24	38.7	39.26

TABLE IV

Accuracy and time comparisons between Multiclass SVM (M-SVM), Gaussian Processes for Ordinal Regression (GPOR), and MSRSVM are provided in the task of estimating 85 ordinal attribute variables. Average mean absolute errors and its standard deviation σ_e are obtained over all attributes. The average training and testing times per attribute are reported along with the standard deviations σ_t.

Method	**Average Mean Absolute Errors (σ_e)**	**Training and Testing Time in secs (σ_t)**
M-SVM	0.6764(0.4985)	40.2286(21.96)
GPOR	0.4816(0.4382)	246.41(215.22)
MSRSVM	0.4706(0.3972)	61.2927(9.6012)

TABLE V

Comparison of the mean absolute error rate of the proposed ranking algorithms, and Support Vector Regression.

Dataset	train/test (d)	ranking (R)	Methods		
			MRSVM	MSRSVM	SVR
Animals with Attributes	3680/920 (2688)	1 (4)	1.08	1.08	1.15
		2 (4)	0.98	1.18	1.5
		avg	1.03	1.13	1.32

TABLE VI

Shown here are the names of the animals in the testing set that have the most number of samples ranked at position (*i*, *j*) estimated with the ranking function learned with MRSVM. The number of samples out of 92/animal associated to a ranking are shown in parenthesis. There are no animals associated with the empty cells.

		Whiteness			
		1	**2**	**3**	**4**
Brownness	1	-	humpback +whale (21)	-	-
	2	leopard (3)	-	-	raccoon (2)
	3	hippopotamus (10)	rat (50)	-	giant +panda (8)
	4	chimpanzee (4)	pig (23)	persian +cat (2)	-