An O(n log n) Cutting Plane Algorithm for Structured Output Ranking

Matthew Blaschko, Arpit Mittal, Esa Rahtu

An $\mathcal{O}(n \log n)$ Cutting Plane Algorithm for Structured Output Ranking

Matthew B. Blaschko[1], Arpit Mittal[2], and Esa Rahtu[3]

[1] Inria & École Centrale Paris, France
[2] Department of Engineering Science, University of Oxford, United Kingdom
[3] Center for Machine Vision Research, University of Oulu, Finland

Abstract. In this work, we consider ranking as a training strategy for structured output prediction. Recent work has begun to explore structured output prediction in the ranking setting, but has mostly focused on the special case of bipartite preference graphs. The bipartite special case is computationally efficient as there exists a linear time cutting plane training strategy for hinge loss bounded regularized risk, but it is unclear how to feasibly extend the approach to complete preference graphs. We develop here a highly parallelizable $\mathcal{O}(n \log n)$ algorithm for cutting plane training with complete preference graphs that is scalable to millions of samples on a single core. We explore theoretically and empirically the relationship between slack rescaling and margin rescaling variants of the hinge loss bound to structured losses, showing that the slack rescaling variant has better stability properties and empirical performance with no additional computational cost per cutting plane iteration. We further show generalization bounds based on uniform convergence. Finally, we demonstrate the effectiveness of the proposed family of approaches on the problem of object detection in computer vision.

1 Introduction

Learning to rank is a core task in machine learning and information retrieval [13]. We consider here a generalization to structured output prediction of the pairwise ranking SVM introduced in [9]. Similar extensions of ranking to the structured output setting [3] have recently been explored in [5, 16, 22]. In these works, pairwise constraints were introduced between elements in a structured output space, enforcing a margin between a lower ranked item and a higher ranked item proportional to the difference in their structured output losses. These works consider only bipartite preference graphs. Although efficient algorithms exist for cutting plane training in the bipartite special case, no feasible algorithm has previously been proposed for extending this approach to fully connected preference graphs for arbitrary loss functions.

Our work makes feasible structured output ranking with a complete preference graph for arbitrary loss functions. Joachims previously proposed an algorithm for ordinal regression with 0-1 loss and R possible ranks in $\mathcal{O}(nR)$ time for n samples [10]. This effectively enables a complete preference graph in this special setting. In practice, however, for structured output prediction with a sufficiently rich output space, the loss values may not be discrete, and may grow linearly with the number of samples. In this case, R is $\mathcal{O}(n)$. Mittal et al. have extended Joachims' $\mathcal{O}(nR)$ result to the structured output ranking setting in the case that there are a discrete set of loss values [15]. A direct extension of these approaches to the structured output setting with a fully connected preference graph and arbitrary loss functions results in a $\mathcal{O}(n^2)$ cutting plane iteration. One of the key contributions of our work is to show that this can be improved to $\mathcal{O}(n \log n)$ time. This enables us to train an objective with 5×10^7 samples on standard hardware (Section 5). Furthermore, straightforward parallelization schemes enable e.g. $\mathcal{O}(n)$ computation time on $\mathcal{O}(\log n)$ processors (Section 3.1). These results hold not only for the structured output prediction setting, but can be used to improve the computational efficiency of related ranking SVM approaches, e.g. [10].

Analogous to the structured output SVM [17, 18], we formulate structured output ranking in slack rescaling and margin rescaling variants. We show uniform convergence bounds for our ranking objective in a unified setting for both variants. Interestingly, the bounds for slack rescaling are dependent on the range of the loss values, while those for margin rescaling are not. Further details are given in Section 4. Structured output ranking is a natural strategy for cascade learning, in which an inexpensive feature function, ϕ, is used to filter a set of possible outputs y. We show empirical results in the cascade setting (Section 5) supporting the efficiency, accuracy, and generalization of the proposed solution to structured output prediction.

2 Structured Output Ranking

The setting considered here is to learn a compatibility function $g : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ that maps an input-output tuple to a real value indicating the prediction of how suitable the input is to a given output. We assume that there is an underlying ground truth prediction for a given input so that every $x_i \in \mathcal{X}$ in a training set is associated with a y_i^* corresponding to the optimal prediction for that input. Additionally, we assume that a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ is provided that measures the similarity of a hypothesized output to the optimal prediction $\Delta(y_i^*, y) \geq 0$. A training set will consist of input-ground truth-output tuples, where the input-ground truth pairs may be repeated, and the outputs are sampled over the output space: $\mathcal{S} = \{(x_i, y_i^*, y_i)\}_{1 \leq i \leq n}$ and (x_i, y_i^*) may equal (x_j, y_j^*) for $j \neq i$ (cf. Section 5). We will use the notation Δ_i to denote $\Delta(y_i^*, y_i)$.

In structured output ranking, we minimize with respect to a compatibility function, g, a risk of the form [1]

$$\begin{aligned} R(g) = \mathbb{E}_{((X_i,Y_i),(X_j,Y_j))}\big[|\Delta_{Y_j} - \Delta_{Y_i}| \cdot \big(\mathbf{1}((\Delta_{Y_j} - \Delta_{Y_i})(g(X_i,Y_i) - g(X_j,Y_j)) < 0) \\ + \frac{1}{2}\mathbf{1}(g(X_i,Y_i) = g(X_j,Y_j))\big)\big], \end{aligned} \tag{1}$$

where $\mathbf{1}(\cdot)$ evaluates to 1 if the argument is true and 0 otherwise, and the term penalizing equality is multiplied by $\frac{1}{2}$ in order to avoid double counting the penalty over the expectation. Here Δ_{Y_i} is the structured output loss associated with an output, Y_i. In contrast to other notions of risk, we take the expectation not with respect to a single sample, but with respect to pairs indexed by the structured output. Given two possible outputs sampled from some prior, the risk determines whether the samples are properly ordered according to the loss associated with predicting that output, and if not pays a penalty proportional to the difference in the losses. This risk penalizes pairs for which sample i has lower loss than sample j and also lower ranking score, i.e. we would like elements with low loss to be ranked higher than elements with high loss.

Two piecewise linear convex upper bounds are commonly used in structured output prediction: a margin rescaled hinge loss, and a slack rescaled hinge loss. The structured output ranking objectives corresponding to regularized risk minimization with these choices are

$$\min_{w \in \mathcal{H}, \xi \in \mathbb{R}} \quad \lambda \Omega(w) + \xi \tag{2}$$

$$\text{s.t.} \quad \sum_{(i,j)\in\mathcal{E}} \nu_{ij} \overbrace{\left(\langle w, \phi(x_i, y_i)\rangle - \langle w, \phi(x_j, y_j)\rangle + \Delta_i - \Delta_j\right)}^{\text{margin rescaling}} \geq -\xi \tag{3}$$

$$\text{or} \quad \sum_{(i,j)\in\mathcal{E}} \nu_{ij} \underbrace{\left(\langle w, \phi(x_i, y_i) - \phi(x_j, y_j)\rangle - 1\right)\left(\Delta_j - \Delta_i\right)}_{\text{slack rescaling}} \geq -\xi \tag{4}$$

$$\xi \geq 0 \qquad \forall \nu \in \{0,1\}^{|\mathcal{E}|} \tag{5}$$

where $\mathcal{E}$ is the edge set associated with a preference graph $\mathcal{G}$,[4] and Ω is a regularizer monotonically increasing in some function norm applied to w [12]. We have presented the one-slack variant here [11]. For a finite sample of (x_i, y_i^*, y_i), such objectives can be solved using a cutting plane approach [10, 15, 16, 20].

The form of $\mathcal{G}$ defines risk variants that encode different preferences in ranking. If an edge exists from node i to node j, this indicates that i should be ranked higher than j. Of particular interest in this work are bipartite graphs, which have efficiencies in computation, and fully connected graphs, which attempt to enforce a total ordering on the samples. Structured output ranking with bipartite preference graphs was previously explored in [16], in which a linear time algorithm was presented for a cutting plane iteration. The algorithm presented in that work shares key similarities with previous strategies for cutting plane training of ranking support vector machines [10], but extends the setting to rescaled structured output losses. A linear time algorithm for fully connected preference graphs was presented in [15] in the special case that the loss values are in a small discrete set. Previous algorithms all degenerate to $\mathcal{O}(n^2)$ when applied to fully connected preference graphs with arbitrary loss values.

[4] An edge from i to j in $\mathcal{G}$ indicates that output i should be ranked above output j. It will generally be the case that $\Delta_j \geq \Delta_i$ for all $(i, j) \in \mathcal{E}$.

3 $\mathcal{O}(n \log n)$ Cutting Plane Algorithm

Algorithm 1 Finding maximally violated slack-rescaled constraint for structured output ranking with a complete bipartite preference graph.

```
Input: $\Delta$, a list of loss values sorted from lowest to highest; $s$, a vector of the current estimate of compatibility scores ($s_u = \langle w, \phi(x_u, y_u)\rangle_{\mathcal{H}}$) in the same order as $\Delta$; $p$, a vector of indices such that $s_{p_v} > s_{p_u}$ whenever $v > u$; $t$, a threshold such that $(u, v) \in \mathcal{E}$ whenever $u \leq t$ and $v > t$
Output: Maximally violated constraint is $\delta - \langle w, \sum_i \alpha_i \phi(x_i, y_i)\rangle \leq \xi$
$p^+ = p_{\{u|p_u \leq t\}}$, $p^- = p_{\{v|p_v > t\}}$
$i = 1$, $\delta = \Delta_+ = 0$, $\Delta^{\text{cum}} = \mathbf{0}$, $\alpha = \mathbf{0}$
$\Delta^{\text{cum}}_{n-t} = \Delta_{p^-_{n-t}}$
for $k = n - t - 1$ to 1 descending do
    $\Delta^{\text{cum}}_{p^-_k} = \Delta_{p^-_k} + \Delta^{\text{cum}}_{p^-_{k+1}}$
end for
for $j = 1$ to $n - t$ do
    while $s_{p^-_j} + 1 > s_{p^+_i} \wedge i \leq t + 1$ do
        $\alpha_{p^+_i} = \alpha_{p^+_i} + \Delta^{\text{cum}}_{p^-_j} - (n - t - j + 1)\Delta_{p^+_i}$
        $\Delta_+ = \Delta_+ + \Delta_{p^+_i}$, $i = i + 1$
    end while
    $\alpha_{p^-_j} = \alpha_{p^-_j} - ((j - 1)\Delta_{p^-_j} - \Delta_+)$
    $\delta = \delta + (j - 1)\Delta_{p^-_j} - \Delta_+$
end for
return $(\alpha, \delta)$
```

Algorithm 2 An $\mathcal{O}(n \log n)$ recursive algorithm for computing a cutting plane iteration for fully connected ranking preference graphs.

```
Input: $\Delta$, a list of loss values sorted from lowest to highest; $s$, a vector of the current estimate of compatibility scores ($s_u = \langle w, \phi(x_u, y_u)\rangle_{\mathcal{H}}$) in the same order as $\Delta$; $p$, an index such that $s_{p_v} > s_{p_u}$ whenever $v > u$
Output: Maximally violated constraint is $\delta - \langle w, \sum_i \alpha_i \phi(x_i, y_i)\rangle \leq \xi$
$n = \text{length}(\Delta)$
if $\Delta_1 = \Delta_n$ then
    return $(\mathbf{0}, 0)$
end if
$t \approx \frac{n}{2}$
$p^a = p_{\{u|p_u \leq t\}}$
$(\alpha_1, \delta_1) = \text{Algorithm } 2(\Delta_{1:t}, s_{1:t}, p^a)$
$p^b = p_{\{v|p_v > t\}}$
$p^b = p^b - t$ (subtract $t$ from each element of $p^b$)
$(\alpha_2, \delta_2) = \text{Algorithm } 2(\Delta_{t+1:n}, s_{t+1:n}, p^b)$
$(\alpha_0, \delta_0) = \text{Algorithm } 1(\Delta, s, p, t)$
$\alpha = \alpha_0 + \alpha_1 + \alpha_2$, $\delta = \delta_0 + \delta_1 + \delta_2$
return $(\alpha, \delta)$
```

Cutting plane optimization of (2)-(5) consists of alternating between optimizing the objective with a finite set of active constraints, finding a maximally violated constraint of the current function estimate and adding it to the active constraint set [11]. Algorithm 1 gives a linear time procedure for finding the maximally violated constraint in the case of a complete bipartite preference graph [10, 16] and slack rescaling.[5] This algorithm follows closely the ordinal regression cutting plane algorithm of [10], and works by performing an initial sort on the current estimate of the sample scores. The algorithm subsequently makes use of the transitivity of violated pairwise constraints to sum all violated pairs in a single pass through the sorted list of samples.

In the case of fully connected preference graphs, Algorithm 2 is a recursive function that ensures that all pairs of samples are considered. Algorithm 2 uses

[5] An analogous algorithm for margin rescaling was given in [16] and has the same computational complexity.

a divide and conquer strategy and works by repeatedly calling Algorithm 1 for various bipartite subgraphs with disjoint edge sets, ensuring that the union of the edge sets of all bipartite subgraphs is the edge set of the preference graph. The set of bipartite subgraphs is constructed by partitioning the set of samples into two roughly equal parts by thresholding the loss function. As the samples are assumed to be sorted by their structured output loss, we simply divide the set by computing the index of the median element. In the event that there are multiple samples with the same loss, the partitioning (Algorithm 2, line 5) may do a linear time search from the median loss value to find a partitioning of the samples such that the first set has strictly lower loss than the second. The notation $p^a = p_{\{u|p_u \leq t\}}$ indicates that p^a contains the elements satisfying the condition in the subscript in the same order that they occured in p. Source code is available for download.[6]

3.1 Complexity

Prior to calling either of the algorithms, the current data sample must be sorted by its structured output loss. Additionally an index vector, p, must be computed that encodes a permutation matrix that sorts the training sample by the current estimate of its compatibility scores, $\langle w, \phi_i \rangle$. Each of these operations has complexity $\mathcal{O}(n \log n)$. The serial complexity of computing the most violated 1-slack constraint is $\mathcal{O}(n \log_2 n)$, matching the complexity of the sorting operation. To show this, we consider the recursion in Algorithm 2. The computational costs of each call consist of (i) the processing needed to find the sorted list of scores for the higher ranked and lower ranked subsets in the bipartite graph, (ii) the cost of calling Algorithm 1, and (iii) the cost of recursion. We will show that items (i) and (ii) can be computed in time linear in the number of samples.

That item (i) is linear in its complexity can be seen by noting that an index p already exists to sort the complete data sample. Rather than pay $\mathcal{O}(n \log n)$ to re-sort the subsets of samples, we may iterate through the elements of p once. As we do so, if $p_i \leq t$, we may add this element to the index that sorts the higher ranked subset. If $p_j > t$, we may add $p_j - t$ to the index that sorts the lower ranked subset. Item (ii) is also linear as the algorithm loops once over each data sample, executing a constant number of operations each time.

We calculate the complexity of Algorithm 2 by a recursive formula $R_n = C_n + 2R_{\frac{n}{2}}$ where C_n is the $\mathcal{O}(n)$ cost of processing items (i) and (ii). It follows that

$$R_n = \sum_{i=0}^{\log_2 n} C_{\frac{n}{2^i}} 2^i. \tag{6}$$

Examining the term $C_{\frac{n}{2^i}} 2^i$, we note that $C_{\frac{n}{2^i}}$ is $\mathcal{O}(\frac{n}{2^i})$ and must be paid 2^i times, resulting in a cost of $\mathcal{O}(n)$ per summand. As there are $\mathcal{O}(\log_2 n)$ summands, the total cost is $\mathcal{O}(n \log n)$. Graphically, the recursion tree is a binary tree in which the cost of each node is proportional to $\frac{1}{2^d}$, where d is the depth of the node

[6] http://pages.saclay.inria.fr/matthew.blaschko/projects/structrank/

(Figure 1). A C implementation of the algorithm takes a fraction second for 10^5 samples on a 2.13 GHz processor.

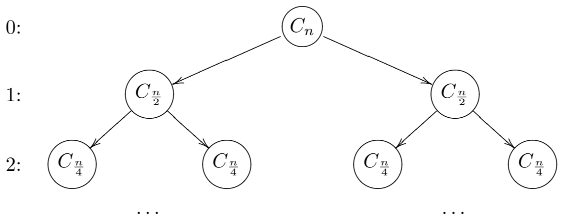


Fig. 1. The recursion tree for Algorithm 2. Each node in the tree corresponds to a set of constraints resulting from a bipartite preference graph. The cost of computing these constraints is labeled in each of the nodes.

A straightforward parallelization scheme can be achieved by placing each recursive call in its own thread. Doing so results in $\mathcal{O}(n)$ computation on $\mathcal{O}(\log n)$ processors: each level of a tree at depth i can be computed independently in $C_{\frac{n}{2^i}} 2^i$ instructions, and there are $\mathcal{O}(\log n)$ levels of the tree. Each of $\log n$ processors can be assigned the nodes corresponding to a given level of the tree.

4 Generalization Bounds

In this section, we develop generalization bounds based on the uniform convergence bounds for ranking algorithms presented in [1]. For $\varDelta \in [0, 1)$ we have tighter bounds for slack rescaling as compared to margin rescaling. For $\varDelta \in [0, \sigma]$ where $\sigma > 1$ bounds are tighter for margin rescaling.

Definition 1 (Uniform loss stability (β)). *A ranking algorithm which is trained on the sample $\mathcal{S}$ of size n has a uniform loss stability β with respect to the ranking loss function ℓ if,*

$$|\ell(\mathcal{S}) - \ell(\mathcal{S}^k)| \leq \beta(n), \ \forall n \in \mathbb{N}, 1 \leq k \leq n \tag{7}$$

where $\mathcal{S}^k$ is a sample resulting from changing the kth element of $\mathcal{S}$, i.e., changing the input training sample by a single example leads to a difference of at most $\beta(n)$ in the loss incurred by the output ranking function on any pair of examples. Thus, a smaller value of $\beta(n)$ corresponds to a greater loss stability.

Definition 2 (Uniform score stability (ν)). *A ranking algorithm with an output $g_{\mathcal{S}}$ on the training sample $\mathcal{S}$ of size n, has a uniform score stability ν if*

$$|g_{\mathcal{S}}(x) - g_{\mathcal{S}^k}(x)| \leq \nu(n), \ \forall\, n \in N, 1 \leq k \leq n, \forall x \in \mathcal{X} \tag{8}$$

i.e., changing an input training sample by a single example leads to a difference of at most $\nu(n)$ in the score assigned by the ranking function to any instance x.

The hinge losses for margin and slack rescaling formulations are given by:

$$\ell_m = (|\varDelta_j - \varDelta_i| - \langle w, \phi(x_i, y_i) - \phi(x_j, y_j)\rangle \cdot \operatorname{sign}(\varDelta_j - \varDelta_i))_+, \tag{9}$$

$$\ell_s = (|\varDelta_j - \varDelta_i| \cdot (1 - \langle w, \phi(x_i, y_i) - \phi(x_j, y_j)\rangle \cdot \operatorname{sign}(\varDelta_j - \varDelta_i)))_+. \tag{10}$$

Theorem 1. *Let $\mathcal{A}$ be a ranking algorithm whose output on a training sample $\mathcal{S} \in (\mathcal{X}, \mathcal{Y})^n$ we denote by $f_{\mathcal{S}}$. Let $\nu : \mathbb{N} \to \mathbb{R}$ be such that $\mathcal{A}$ has uniform score stability ν. $\mathcal{A}$ has uniform loss stability β with respect to the slack rescaling loss ℓ_s, where for all $n \in \mathbb{N}$*

$$\beta(n) = 2\sigma\nu(n) \tag{11}$$

where $\sigma \geq \Delta$ is an upper bound on the structured output loss function.

Proof. Without loss of generality we assume that $\ell_s(\mathcal{S}) > \ell_s(\mathcal{S}^k)$. There are two non-trivial cases.

Case (i): Margin is violated by both $g_{\mathcal{S}}$ and $g_{\mathcal{S}^k}$.

$$\begin{aligned} |\ell_s(\mathcal{S}) - \ell_s(\mathcal{S}^k)| =& |\Delta_j - \Delta_i| \cdot (1 - (g_{\mathcal{S}}(x_i) - g_{\mathcal{S}}(x_j)) \cdot \mathrm{sign}(\Delta_j - \Delta_i)) - & (12)\\ & |\Delta_j - \Delta_i| \cdot (1 - (g_{\mathcal{S}^k}(x_i) - g_{\mathcal{S}^k}(x_j)) \cdot \mathrm{sign}(\Delta_j - \Delta_i)) \\ \leq& \sigma(|g_{\mathcal{S}}(x_i) - g_{\mathcal{S}^k}(x_i)| + |g_{\mathcal{S}}(x_j) - g_{\mathcal{S}^k}(x_j)|) \leq 2\sigma\nu(n) & (13)\end{aligned}$$

Case (ii): Margin is violated by either of $g_{\mathcal{S}}$ or $g_{\mathcal{S}^k}$. This is a symmetric case, so we assume that the margin is violated by $g_{\mathcal{S}}$.

$$\begin{aligned} |\ell_s(\mathcal{S}) - \ell_s(\mathcal{S}^k)| =& |\Delta_j - \Delta_i| \cdot (1 - (g_{\mathcal{S}}(x_i) - g_{\mathcal{S}}(x_j)) \cdot \mathrm{sign}(\Delta_j - \Delta_i)) & (14)\\ \leq& |\Delta_j - \Delta_i| \cdot (1 - (g_{\mathcal{S}}(x_i) - g_{\mathcal{S}}(x_j)) \cdot \mathrm{sign}(\Delta_j - \Delta_i)) - & (15)\\ & |\Delta_j - \Delta_i| \cdot (1 - (g_{\mathcal{S}^k}(x_i) - g_{\mathcal{S}^k}(x_j)) \cdot \mathrm{sign}(\Delta_j - \Delta_i)) \\ \leq& \sigma(|g_{\mathcal{S}}(x_i) - g_{\mathcal{S}^k}(x_i)| + |g_{\mathcal{S}}(x_j) - g_{\mathcal{S}^k}(x_j)|) \leq 2\sigma\nu(n) & (16)\end{aligned}$$

Theorem 2 (Slack Rescaling Generalization Bound). *Let $\mathcal{H}$ be a RKHS with a joint-kernel[7] k such that $\forall(x, y) \in \mathcal{X} \times \mathcal{Y}$, $k((x, y),(x, y)) \leq \kappa^2 < \infty$. Let $\lambda > 0$ and ℓ_r be a rescaled ramp loss. The training algorithm trained on sample $\mathcal{S}$ of size n outputs a ranking function $g_{\mathcal{S}} \in \mathcal{H}$ that satisfies $g_{\mathcal{S}} = \arg\min_{g \in \mathcal{H}}\{\hat{R}_{\ell_s}(g; \mathcal{S}) + \lambda||g||^2_{\mathcal{H}}\}$. Then for any $0 < \delta < 1$, with probability at least $1 - \delta$ over the draw of $\mathcal{S}$, the expected ranking error of the function is bounded by:*

$$R(g_{\mathcal{S}}) < \hat{R}_{\ell_r}(g_{\mathcal{S}}; \mathcal{S}) + \frac{32\sigma^2\kappa^2}{\lambda n} + \left(\frac{16\sigma^2\kappa^2}{\lambda} + \sigma\right)\sqrt{\frac{2\ln(1/\delta)}{n}} \tag{17}$$

Proof. From [1, Theorem 11], $\nu(n) = \frac{8\sigma\kappa^2}{\lambda n}$. Substituting this value of $\nu(n)$ in Equation (11) $\beta(n) = \frac{16\sigma^2\kappa^2}{\lambda n}$. Inequality (17) then follows by an application of [1, Theorem 6] which gives the generalization bound as a function of $\beta(n)$.

The proof of [1, Theorem 6] follows closely that of [6] for regression and classification, relying at its core on McDiarmid's inequality [14].

Theorem 3 (Margin Rescaling Generalization Bound). *Under the conditions of Theorem 2, and a ranking function $f_{\mathcal{S}} \in \mathcal{H}$ that satisfies $f_{\mathcal{S}} = \arg\min_{f \in \mathcal{H}}\{\hat{R}_{\ell_m}(f; \mathcal{S}) + \lambda||f||^2_{\mathcal{H}}\}$. Then for any $0 < \delta < 1$, with probability*

[7] We assume a joint kernel map of the form given in [17, 18].

at least $1 - \delta$ *over the draw of* $\mathcal{S}$*, the expected ranking error of the function is bounded by:*

$$R(f_{\mathcal{S}}) < \hat{R}_{\ell_r}(f_{\mathcal{S}}; \mathcal{S}) + \frac{32\kappa^2}{\lambda n} + \left(\frac{16\kappa^2}{\lambda} + \sigma \right) \sqrt{\frac{2\ln(1/\delta)}{n}} \tag{18}$$

The proof of Theorem 3 follows the outline given in [1, Section 5.2.1].

5 Experimental Results

Results are presented as an evaluation of a cascade architecture [21], following the evaluation protocol of Rahtu et al. [16]. The experiments are presented on the VOC 2007 dataset [7]. The images are annotated with ground-truth bounding boxes of objects from 20 classes. VOC 2007 train and validation sets are used only to construct the distribution for the initial window sampling, and the ranking function is learned using the dataset presented in [2]. This is done in order to obtain results comparable to those in [2, 16].

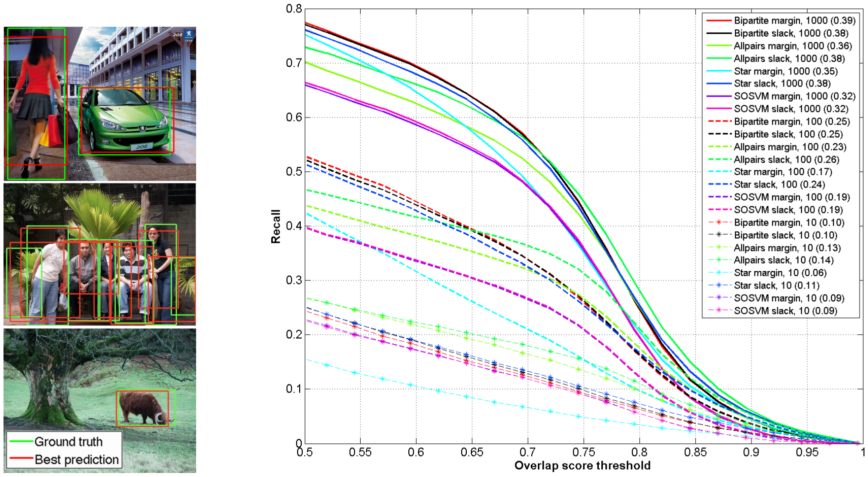
The performance is measured using a recall-overlap curve, which indicates the recall rate of ground truth boxes in the VOC 2007 test set for a given minimum value of the overlap score [19]

$$o(y, \tilde{y}) = \frac{\text{Area}(y \cap \tilde{y})}{\text{Area}(y \cup \tilde{y})}, \tag{19}$$

where y and $\tilde{y}$ denote the ground truth and predicted bounding box, respectively. We also report the area under the curve (AUC) between overlap scores 0.5 and 1, and normalize its value so that the maximum is 1 for perfect recall. The overlap limit 0.5 is chosen here since less accurately localized boxes have little practical importance.

Our framework for creating the set of predicted bounding boxes broadly follows that of [16]. This setting has three main stages: (i) construction of the initial bounding boxes, (ii) feature extraction, and (iii) window selection. In the first stage we generate a pool of approximately 100,000 initial windows per image using random sampling and superpixel bounding boxes. The random samples are drawn from a distribution learned using the ground truth object boxes in the training and validation sets. The superpixels are computed by a graph based method [8], which is selected for its computational efficiency. At overlap 0.5, the initial windows achieve approximately 98% recall.

In the second stage, the tentative bounding boxes are scored using several publicly available features. These features are window symmetry (WS), boundary edge distribution (BE), superpixel boundary integral (BI), color contrast (CC), superpixel straddling (SS), and multiscale saliency (MS). The WS, BE, and BI features are described in [16] and SS, CC, and MS are from [2]. The joint feature map, $\phi(x_i, y_i)$, applied in learning is the feature vector corresponding to the bounding box y_i.



(a) Example detections with a complete preference graph and slack rescaling. This setting corresponds to “Allpairs slack, 1000” in Fig. (b).

(b) Overlap/recall curves. Results are presented for varying preference graphs, margin and slack rescaling, and various numbers of returned windows. The AUC score is given in parentheses (a higher number at a given number of returned windows indicates better performance).

Fig. 2. Example detections and overlap vs. recall for an object detection task. See Section 5 for a complete description of the experimental setting. This figure is best viewed in color.

In the last stage, we select the final set of bounding boxes (10, 100 or 1000) based on the learned score. The feature weights for the linear combination are learned by using the structured output ranking framework presented in this paper and the loss function proposed in [4]. This loss is based on the overlap ratio (19) and is defined as $\Delta_i \equiv 1 - o(y, y_i)$.

In order to run the proposed algorithm, we further need to define the structure of the preference graph $\mathcal{G}$. Three variants were considered: a bipartite graph in which 1000 best samples per image are ranked higher than all other initial windows (as in [16]), a fully connected graph (denoted “Allpairs” in the legend of Figure 2(b)) where full ranking is pursued, or a bipartite graph in which only ground truth windows are to be ranked higher than all sampled windows (denoted “Star” in the legend, as the topology of a bipartite graph with one singleton set is a star graph). Finally, we have trained a standard structured output SVM (labeled “SOSVM”) in the same manor as [4]. To ensure a diverse set of predictions, we have applied the non-maximal suppression approach described in [19].

The overlap-recall curves are shown in Figure 2. The legend in Figure 2 encodes the experimental setting for each curve. First, the structure of the preference graph, $\mathcal{G}$, is specified. The second component of the legend indicates whether slack rescaling or margin rescaling was employed. The third component states the number of top ranked windows used for evaluating the recall. Finally, the fourth component (in parentheses) gives the AUC value.

6 Discussion

The experiments described in Section 5 show that structured output ranking is a natural objective to apply to cascade detection models.

On average, a bipartite preference graph performs best if we require 1000 windows as output, which matches the training conditions. The bipartite graph was constructed such that constraints were included between the top 1000 sampled windows, and the remaining 99000 windows. However, when the number of returned windows deviates from 1000, the relative performance of the bipartite ranking decreases and other preference graphs give better performance. The objective is tuned to give the highest performance under a single evaluation setting, at the expense of other settings.

The complete preference graph ranking, labeled “Allpairs” in Figure 2, gives good performance and tends to have higher performance at high overlap levels. While the difference between slack rescaling and margin rescaling was minimal when using a bipartite preference graph, a much more noticeable difference is present in the case of a complete preference graph. While the bipartite preference graph performs better at certain overlap levels when 1000 windows are returned, the complete preference graph is much more stable across a wide number of windows, and gives the best performance at all overlap levels if 10 windows are returned per image. Finally, the standard structured output SVM (labeled “SOSVM”) performs substantially worse than all ranking variants.

7 Conclusions

In this work, we have explored the use of ranking for structured output prediction. We have analyzed both margin and slack rescaling variants of a ranking SVM style approach, showing better empirical results for slack rescaling, and proving generalization bounds for both variants in a unified framework. Furthermore, we have proposed an efficient and parallelizable algorithm for cutting plane training that scales to millions of data points on a single core. We have shown an example application of object detection in computer vision, demonstrating that ranking methods outperform a standard structured output SVM in this setting, and that fully connected preference graphs give excellent performance across a range of settings, particularly at high overlap with the ground truth.

The $\mathcal{O}(n \log n)$ algorithm presented here can be adapted to a wide variety of settings, improving the computational efficiency in a range of ranking approaches and applications. In the setting of [10, 15], the $\mathcal{O}(nR)$ approach for ranking with a complete preference graph and a fixed number, R, of loss values can be improved in an analogous manner to $\mathcal{O}(n \log R)$.

Acknowledgements
This work is partially funded by ERC Grant 259112, and FP7-MC-CIG 334380.

References

1. Agarwal, S., Niyogi, P.: Generalization bounds for ranking algorithms via algorithmic stability. Journal of Machine Learning Research 10, 441–474 (June 2009)
2. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2010)
3. Bakır, G.H., Hofmann, T., Schölkopf, B., Smola, A.J., Taskar, B., Vishwanathan, S.V.N.: Predicting Structured Data. MIT Press (2007)
4. Blaschko, M.B., Lampert, C.H.: Learning to localize objects with structured output regression. In: Proceedings of the European Conference on Computer Vision (2008)
5. Blaschko, M.B., Vedaldi, A., Zisserman, A.: Simultaneous object detection and ranking with weak supervision. In: Advances in Neural Information Processing Systems (2010)
6. Bousquet, O., Elisseeff, A.: Stability and generalization. Journal of Machine Learning Research 2, 499–526 (2002)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88(2), 303–338 (June 2010)
8. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. International Journal of Computer Vision 59(2), 167–181 (2004)
9. Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. In: Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers. pp. 115–132. MIT Press (2000)
10. Joachims, T.: Training linear SVMs in linear time. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 217–226 (2006)
11. Joachims, T., Finley, T., Yu, C.N.: Cutting-plane training of structural SVMs. Machine Learning 77(1), 27–59 (2009)
12. Lafferty, J., Zhu, X., Liu, Y.: Kernel conditional random fields: representation and clique selection. In: Proceedings of the International Conference on Machine Learning (2004)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
14. McDiarmid, C.: On the method of bounded differences. In: Siemons, J. (ed.) Surveys in Combinatorics. pp. 148–188. Cambridge University Press (1989)
15. Mittal, A., Blaschko, M.B., Zisserman, A., Torr, P.H.S.: Taxonomic multi-class prediction and person layout using efficient structured ranking. In: Proceedings of the European Conference on Computer Vision (2012)
16. Rahtu, E., Kannala, J., Blaschko, M.B.: Learning a category independent object detection cascade. In: Proceedings of the International Conference on Computer Vision (2011)
17. Taskar, B., Guestrin, C., Koller, D.: Max-margin Markov networks. In: Advances in Neural Information Processing Systems (2004)
18. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the International Conference on Machine Learning (2004)
19. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: Proceedings of the International Conference on Computer Vision (2009)
20. Vedaldi, A., Blaschko, M.B., Zisserman, A.: Learning equivariant structured output SVM regressors. In: Proceedings of the International Conference on Computer Vision. pp. 959–966 (2011)

21. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision 57(2), 137–154 (2002)
22. Zhang, Z., Warrell, J., Torr, P.: Proposal generation for object detection using cascaded ranking SVMs. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2011)