Journal of Machine Learning Research ? (2005) ? Submitted ?; Published ?

Efficient Learning of Label Ranking by Soft Projections onto Polyhedra

Shai Shalev-Shwartz* SHAIS@CS.HUJI.AC.IL
School of Computer Science and Engineering
The Hebrew University
Jerusalem 91904, Israel

Yoram Singer† SINGER@GOOGLE.COM
Google Inc.
1600 Amphitheatre Pkwy
Mountain View, CA, 94043

Editor: Kristin Bennett and Emilio Parrado-Hernandez

Abstract

We discuss the problem of learning to rank labels from a real valued feedback associated with each label. We cast the feedback as a preferences graph where the nodes of the graph are the labels and edges express preferences over labels. We tackle the learning problem by defining a loss function for comparing a predicted graph with a feedback graph. This loss is materialized by decomposing the feedback graph into bipartite sub-graphs. We then adopt the maximum-margin framework which leads to a quadratic optimization problem with linear constraints. While the size of the problem grows quadratically with the number of the nodes in the feedback graph, we derive a problem of a significantly smaller size and prove that it attains the same minimum. We then describe an efficient algorithm, called SOPOPO, for solving the reduced problem by employing a soft projection onto the polyhedron defined by a reduced set of constraints. We also describe and analyze a wrapper procedure for batch learning when multiple graphs are provided for training. We conclude with a set of experiments which show significant improvements in run time over a state of the art interior-point algorithm.

1. Introduction

To motivate the problem discussed in this paper let us consider the following application. Many news feeds such as Reuters and Associated Press tag each news article they handle with labels drawn from a predefined set of possible topics. These tags are used for routing articles to different targets and clients. Each tag may also be associated with a degree of relevance, often expressed as a numerical value, which reflects to what extent a topic is relevant to the news article on hand. Tagging each individual article is clearly a laborious and time consuming task. In this paper we describe and analyze an efficient algorithmic framework for learning and inferring preferences over labels. Furthermore, in addition to the task described above, our learning apparatus includes as special cases problems ranging from binary classification to total order prediction.

∗. Part of S. Shalev-Shwartz's work was done while visiting Google.
†. Author for correspondences.

We focus on batch learning in which the learning algorithm receives a set of training examples, each example consists of an instance and a target vector. The goal of the learning process is to deduce an accurate mapping from the instance space to the target space. The target space $\mathcal{Y}$ is a *predefined* set of labels. For concreteness, we assume that $\mathcal{Y} = \{1, 2, \ldots, k\}$. The prediction task is to assert preferences over the labels. This setting in particular generalizes the notion of a single tag or label $y \in \mathcal{Y} = \{1, 2, \ldots, k\}$, typically used in multiclass categorization tasks, to a full set of preferences over the labels. Preferences are encoded by a vector $\boldsymbol{\gamma} \in \mathbb{R}^k$, where $\gamma_y > \gamma_{y'}$ means that label y is more relevant to the instance than label y'. The preferences over the labels can also be described as a weighted directed graph: the nodes of the graph are the labels and weighted edges encode pairwise preferences over pairs of labels. In Fig. 1 we give the graph representation for the target vector $(-1, 0, 2, 0, -1)$ where each edge marked with its weight. For instance, the weight of the edge $(3, 1)$ is $\gamma_3 - \gamma_1 = 3$.

The class of mappings we employ in this paper is the set of linear functions. While this function class may seem restrictive, the pioneering work of Vapnik (1998) and colleagues demonstrates that by using Mercer kernels one can employ highly non-linear predictors, called support vector machines (SVM) and still entertain all the formal properties and simplicity of linear predictors. We propose a SVM-like learning paradigm for predicting the preferences over labels. We generalize the definition of the hinge-loss used in SVM to the label ranking setting. Our generalized hinge loss contrasts the predicted preferences graph and the target preferences graph by decomposing the target graph into bipartite sub-graphs. As we discuss in the next section, this decomposition into sub-graphs is rather flexible and enables us to analyze several previously defined loss functions in a single unified setting. This definition of the generalized hinge loss lets us pose the learning problem as a quadratic optimization problem while the structured decomposition leads to an efficient and effective optimization procedure.

The main building block of our optimization procedure is an algorithm which performs fast and frugal SOft Projections Onto a POlyhedron and is therefore abbreviated SOPOPO. Generalizing the iterative algorithm proposed by Hildreth (1957) (see also Censor and Zenios (1997)) from half-space constraints to polyhedra constraints, we also derive and analyze an iterative algorithm which on each iteration performs a soft projection onto a single polyhedron. The end result is a fast optimization procedure for label ranking from general real-valued feedback.

The paper is organized as follows. In Sec. 2 we start with a formal definition of our setting and cast the learning task as a quadratic programming problem. We also make references to previous work on related problems that are covered by our setting. Our efficient optimization procedure for the resulting quadratic problem is described in two steps. First, we present in Sec. 3 the SOPOPO algorithm for projecting onto a single polyhedron. Then, in Sec. 4, we derive and analyze an iterative algorithm which solves the original quadratic optimization problem by successive activations of SOPOPO. Experiments are provided in Sec. 5 and concluding remarks are given in Sec. 6.

Before moving to the specifics, we would like to stress that while the learning task discussed in this paper is well rooted in the machine learning community, the focus of the paper is the design and analysis of an optimization apparatus. The readers interested in the broad problem of learning preferences, including its learning theoretic facets such as generalization properties are referred for instance to (Cohen et al., 1999, Herbrich et al., 2000, Rudin et al., 2005, Agarwal and Niyogi, 2005, Clemenon et al., 2005) and the many references therein.

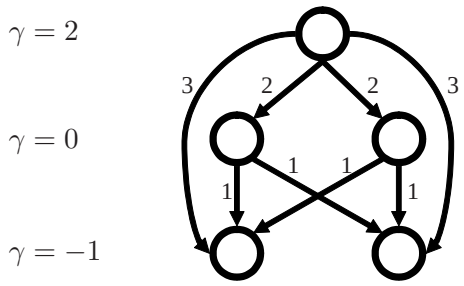

Figure 1: The graph induced by the feedback $\boldsymbol{\gamma} = (-1, 0, 2, 0, -1)$.

2. Problem Setting

In this section we introduce the notation used throughout the paper and formally describe our problem setting. We denote scalars with lower case letters (e.g. x and α), and vectors with bold face letters (e.g. $\mathbf{x}$ and $\boldsymbol{\alpha}$). Sets are designated by upper case Latin letters (e.g. E) and set of sets by bold face (e.g. $\mathbf{E}$). The set of non-negative real numbers is denoted by $\mathbb{R}_+$. For any $k \geq 1$, the set of integers $\{1, \ldots, k\}$ is denoted by $[k]$. We use the notation $(a)_+$ to denote the hinge function, namely, $(a)_+ = \max\{0, a\}$.

Let $\mathcal{X}$ be an instance domain and let $\mathcal{Y} = [k]$ be a predefined set of labels. A target for an instance $\mathbf{x} \in \mathcal{X}$ is a vector $\boldsymbol{\gamma} \in \mathbb{R}^k$ where $\gamma_y > \gamma_{y'}$ means that y is more relevant to $\mathbf{x}$ than y'. We also refer to $\boldsymbol{\gamma}$ as a label ranking. We would like to emphasize that two different labels may attain the same rank, that is, $\gamma_y = \gamma_{y'}$ while $y \neq y'$. In this case, we say that y and y' are of equal relevance to $\mathbf{x}$. We can also describe $\boldsymbol{\gamma}$ as a weighted directed graph. The nodes of the graph are labeled by the elements of $[k]$ and there is a directed edge of weight $\gamma_r - \gamma_s$ from node r to node s iff $\gamma_r > \gamma_s$. In Fig. 1 we give the graph representation for the label-ranking vector $\boldsymbol{\gamma} = (-1, 0, 2, 0, -1)$.

The learning goal is to learn a ranking function of the form $\mathbf{f} : \mathcal{X} \to \mathbb{R}^k$ which takes $\mathbf{x}$ as an input instance and returns a ranking vector $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^k$. We denote by $f_r(\mathbf{x})$ the rth element of $\mathbf{f}(\mathbf{x})$. Analogous to the target vector, $\boldsymbol{\gamma}$, we say that label y is more relevant than label y' with respect to the predicted ranking if $f_y(\mathbf{x}) > f_{y'}(\mathbf{x})$. We assume that the label-ranking functions are linear, namely,

$$f_r(\mathbf{x}) = \mathbf{w}_r \cdot \mathbf{x} \ ,$$

where each $\mathbf{w}_r$ is a vector in $\mathbb{R}^n$ and $\mathcal{X} \subseteq \mathbb{R}^n$. As we discuss briefly at the end of Sec. 4, our algorithm can be generalized straightforwardly to non-linear ranking functions by employing Mercer kernels (Vapnik, 1998).

We focus on a batch learning setting in which a training set $S = \{(\mathbf{x}^i, \boldsymbol{\gamma}^i)\}_{i=1}^m$ is provided. Thus, each example consists of an instance $\mathbf{x}^i \in \mathcal{X}$ and a label-ranking $\boldsymbol{\gamma}^i \in \mathbb{R}^k$. The performance of a label-ranking function $\mathbf{f}$ on an example $(\mathbf{x}, \boldsymbol{\gamma})$ is evaluated via a loss function $\ell : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$. Clearly, we want the loss of a predicted ranking to be small if it expresses similar preferences over pairs as the given label-ranking. Moreover, we view the difference $\gamma_r - \gamma_s$ for a pair of labels r and s as an encoding of the importance of the ordering of r ahead of s. That is, the larger this difference is the more we prefer r over s. We view this requirement as a lower bound on the difference between $f_r(\mathbf{x})$ and $f_s(\mathbf{x})$. Formally, for each pair of labels $(r, s) \in \mathcal{Y} \times \mathcal{Y}$ such that $\gamma_r > \gamma_s$, we define the

loss of $\mathbf{f}$ with respect to the pair as,

$$\ell_{r,s}(\mathbf{f}(\mathbf{x}), \boldsymbol{\gamma}) \;=\; ((\gamma_r - \gamma_s) - (f_r(\mathbf{x}) - f_s(\mathbf{x})))_+ \quad . \tag{1}$$

The above definition of loss extends the hinge-loss used in binary classification problems (Vapnik, 1998) to the problem of label-ranking. The loss $\ell_{r,s}$ reflects the amount by which the constraint $f_r(\mathbf{x}) - f_s(\mathbf{x}) \geq \gamma_r - \gamma_s$ is not satisfied. While the construction above is defined for pairs, our goal though is to associate a loss with the *entire* predicted ranking and not a single pair. Thus, we need to combine the individual losses over pairs into one meaningful loss. In this paper we take a rather flexible approach by specifying an apparatus for combining the individual losses over pairs into a single loss. We combine the different pair-based losses into a single loss by grouping the pairs of labels into independent sets each of which is isomorphic to a *complete bipartite* graph. Formally, given a target label-ranking vector $\boldsymbol{\gamma} \in \mathbb{R}^k$, we define $\mathbf{E}(\boldsymbol{\gamma}) = \{E_1, \ldots, E_d\}$ to be a collection of subsets of $\mathcal{Y} \times \mathcal{Y}$. For each $j \in [d]$, define V_j to be the set of labels which support the edges in E_j, that is,

$$V_j = \{y \in \mathcal{Y} : \exists\, r \text{ s.t. } (r, y) \in E_j \vee (y, r) \in E_j\} \quad . \tag{2}$$

We further require that $\mathbf{E}(\boldsymbol{\gamma})$ satisfies the following conditions,

1. For each $j \in [d]$ and for each $(r, s) \in E_j$ we have $\gamma_r > \gamma_s$.
2. For each $i \neq j \in [d]$ we have $E_i \cap E_j = \emptyset$.
3. For each $j \in [d]$, the sub-graph defined by (V_j, E_j) is a complete bipartite graph. That is, there exists two sets A and B, such that $A \cap B = \emptyset$, $V_j = A \cup B$, and $E_j = A \times B$.

In Fig. 2 we illustrate a few possible decompositions into bipartite graphs for a given label-ranking.

The loss of each sub-graph (V_j, E_j) is defined as the maximum over the losses of the pairs belonging to the sub-graph. In order to add some flexibility we also allow different sub-graphs to have different contribution to the loss. We do so by associating a weight σ_j with each sub-graph. The general form of our loss is therefore,

$$\ell(\mathbf{f}(\mathbf{x}), \boldsymbol{\gamma}) \;=\; \sum_{j=1}^{d} \sigma_j \max_{(r,s) \in E_j} \ell_{r,s}(\mathbf{f}(\mathbf{x}), \boldsymbol{\gamma}) \quad , \tag{3}$$

where each $\sigma_j \in \mathbb{R}_+$ is a non-negative weight. The weights σ_j can be used to associate importance values with each sub-graph (V_j, E_j) and to facilitate different notions of losses. For example, in multilabel classification problems, each instance is associated with a set of relevant labels which come from a predefined set $\mathcal{Y}$. The multilabel classification problem is a special case of the label ranking problem discussed in this paper and can be realized by setting $\gamma_r = 1$ if the r'th label is relevant and otherwise defining $\gamma_r = 0$. Thus, the feedback graph itself is of a bipartite form. Its edges are from $A \times B$ where A consists of all the relevant labels and B of the irrelevant ones. If we decide to set $\mathbf{E}(\boldsymbol{\gamma})$ to contain the single set $A \times B$ and define $\sigma_1 = 1$ then $\ell(\mathbf{f}(\mathbf{x}), \boldsymbol{\gamma})$ amounts to the *maximum* value of $\ell_{r,s}$ over pairs of edges in $A \times B$. Thus, the loss of this decomposition distills to the worst loss suffered over all pairs of comparable labels. Alternatively, we can set $\mathbf{E}(\boldsymbol{\gamma})$ to consist of all the sets $\{(r, s)\}$ for each $(r, s) \in A \times B$ and define $\sigma_j = 1/|\mathbf{E}(\boldsymbol{\gamma})|$. In this case the total loss $\ell(\mathbf{f}(\mathbf{x}), \boldsymbol{\gamma})$ is the *average* value of $\ell_{r,s}$ over the edges in $A \times B$. Clearly, one can devise decompositions of $\mathbf{E}(\boldsymbol{\gamma})$ which are neither all pairs of edges nor a singleton including all edges. We

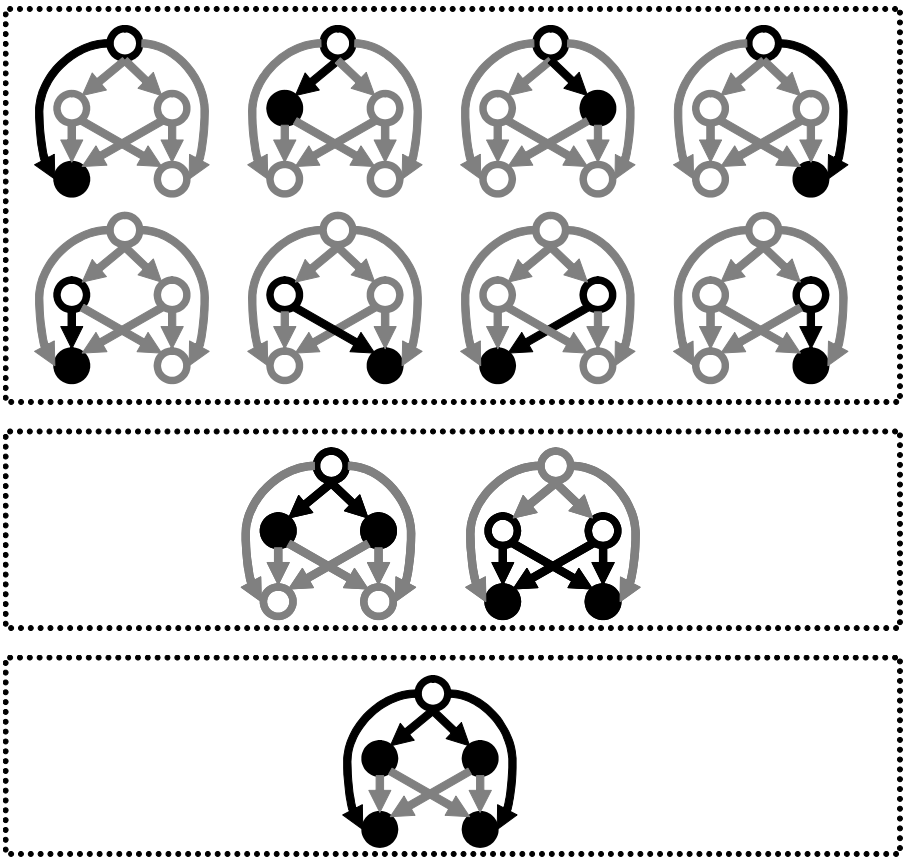

Figure 2: Three possible decompositions into complete bipartite sub-graphs of the graph from Fig. 1. Top: all-pairs decomposition; Middle: all adjacent layers; Bottom: top layer versus the rest of the layers. The edges and vertices participating in each sub-graph are depicted in black while the rest are presented in gray. In each graph the nodes constituting the set A are designated by black circles while for the nodes in B by filled black circles.

can thus capture different notions of losses for label ranking functions with multitude schemes for casting the relative importance of each subset (V_j, E_j).

Equipped with the loss function given in Eq. (3) we now formally define our learning problem. As in most learning settings, we assume that there exists an unknown distribution D over $\mathcal{X} \times \mathbb{R}^k$ and that each example in our training set is identically and independently drawn from D. The ultimate goal is to learn a label ranking function $\mathbf{f}$ which entertains a small generalization loss, $\mathbb{E}_{(\mathbf{x},\boldsymbol{\gamma})\sim D}\left[\ell(\mathbf{f}(\mathbf{x}), \boldsymbol{\gamma})\right]$. Since the distribution is not known we use instead an empirical sample from D and encompass a penalty for excessively complex label-ranking functions. Generalizing the Support Vector Machine (SVM) paradigm, we define a constrained optimization problem, whose optimal solution would constitute our label-ranking function. The objective function we need to minimize is composed of two terms. The first is the empirical loss of the label-ranking function on the training set and the second is a penalty for complexity, often referred to as a regularization term. This term amounts to the sum of the squared norms of $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$. The trade-off between

the regularization term and the empirical loss term is controlled by a parameter C. The resulting optimization problem is,

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_k} \ \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{w}_j\|^2 \ + \ C\sum_{i=1}^{m}\ell(\mathbf{f}(\mathbf{x}^i),\boldsymbol{\gamma}^i) \ , \tag{4}$$

where $f_y(\mathbf{x}^i) = \mathbf{w}_y \cdot \mathbf{x}^i$. Note that the loss function in Eq. (3) can also be represented as the solution of the following optimization problem,

$$\begin{aligned} \ell(\mathbf{f}(\mathbf{x}),\boldsymbol{\gamma}) \ = \ \min_{\boldsymbol{\xi}\in\mathbb{R}^d_+} \ & \sum_{j=1}^{d}\sigma_j\,\xi_j \\ \text{s.t. } \forall j\in[d],\ \forall (r,s)\in E_j, \ & \ \mathbf{f}_r(\mathbf{x}) - \mathbf{f}_s(\mathbf{x}) \geq \gamma_r - \gamma_s - \xi_j \ , \end{aligned} \tag{5}$$

where $d = |\mathbf{E}(\boldsymbol{\gamma})|$. Thus, we can rewrite the optimization problem given in Eq. (4) as a quadratic optimization problem,

$$\begin{aligned} \min_{\mathbf{w}_1,\ldots,\mathbf{w}_k,\boldsymbol{\xi}} \ & \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{w}_j\|^2 \ + \ C\sum_{i=1}^{m}\sum_{j=1}^{|\mathbf{E}(\boldsymbol{\gamma}^i)|}\sigma_j\,\xi_j^i \\ \text{s.t. } & \forall\, i\in[m],\, \forall\, E_j\in\mathbf{E}(\boldsymbol{\gamma}^i),\ \ \forall\,(r,s)\in E_j, \quad \mathbf{w}_r\cdot\mathbf{x}^i - \mathbf{w}_s\cdot\mathbf{x}^i \geq \gamma_r^i - \gamma_s^i - \xi_j^i \\ & \forall\, i,j, \ \ \xi_j^i \geq 0 \ . \end{aligned} \tag{6}$$

To conclude this section, we would like to review the rationale for choosing an one-sided loss for each pair by casting a single inequality for each (r,s). It is fairly easy to define a two-sided loss for a pair by mimicking regression problems. Concretely, we could replace the definition of $\ell_{r,s}$ as given in Eq. (1) with the loss $|f_r(\mathbf{x}) - f_s(\mathbf{x}) - (\gamma_r - \gamma_s)|$. This loss penalizes for *any* deviation from the desired difference of $\gamma_r - \gamma_s$. Instead, our loss is one sided as it penalizes only for not achieving a lower-bound. This choice is more natural in ranking applications. For instance, suppose we need to induce a ranking over 4 labels where the target label ranking is $(-1, 2, 0, 0)$. Assume that the predicted ranking is instead $(-5, 3, 0, 0)$. In most ranking and search applications such a predicted ranking would be perceived as being right on target since the preferences it expresses over pairs are on par with the target ranking. Furthermore, in most ranking applications, overly demotion of the most irrelevant items and excessive promotion of the most relevant ones is perceived as beneficial rather than a deficiency. Put another way, the set of target values encode minimal margin requirements and over-achieving these margin requirements should not be penalized.

Related Work Various known supervised learning problems can be viewed as special cases of the label ranking setting described in this paper. First, note that when there are only two labels we obtain the original constrained optimization of support vector machines for binary classification (Cortes and Vapnik, 1995) with the bias term set to zero. In the binary case, our algorithm reduces to the SOR algorithm described in (Mangasarian and Musicant, 1999). The multiclass problem, in which the target is a single label $y \in \mathcal{Y}$, can also be derived from our setting by defining $\gamma_y = 1$ and $\gamma_r = 0$ for all $r \neq y$. A few special-purpose algorithms have been suggested to solve the multiclass SVM problems. The multiclass version of Weston and Watkins (1999) is obtained by defining $\mathbf{E}(\boldsymbol{\gamma}) = \{\{(y,r)\}\}_{r\neq y}$, that is, each subset consists of a single pair (y,r). The multiclass version

of Crammer and Singer (2001) can be obtained by simply setting $\mathbf{E}(\boldsymbol{\gamma})$ to be a single set containing all the pairs (y, r) for $r \neq y$, namely $\mathbf{E}(\boldsymbol{\gamma}) = \{\{(y,1),\ldots,(y,y-1),(y,y+1),\ldots,(y,k)\}\}$. While the learning algorithms from (Weston and Watkins, 1999) and (Crammer and Singer, 2001) are seemingly different, they can be solved using the same algorithmic infrastructure presented in this paper. Proceeding to more complex decision problems, the task of multilabel classification or ranking is concerned with predicting a set or relevant labels or ranking the labels in accordance to their relevance to the input instance. This problem was studied by several authors (Elisseeff and Weston, 2001, Crammer and Singer, 2002, Dekel et al., 2003). Among these studies, the work of Elisseeff and Weston (2001) is probably the closest to ours yet it is still a derived special case of our setting . Elisseeff and Weston focus on a feedback vector $\boldsymbol{\gamma}$ which constitutes a bipartite graph by itself and define a constrained optimization problem with a *separate* slack variable for each edge in the graph. Formally, each instance $\mathbf{x}$ is associated with a set of relevant labels denoted Y. As discussed in the example above, the multilabel categorization setting can thus be realized by defining $\gamma_r = 1$ for all $r \in Y$ and $\gamma_s = 0$ for all $s \notin Y$. The construction of Elisseeff and Weston can be recovered by defining $\mathbf{E}(\boldsymbol{\gamma}) = \{\{(r,s)\}|\gamma_r > \gamma_s\}$. Our approach is substantially more general as it allows much richer and flexible ways to decompose the multilabel problem as well as more general label ranking problems.

3. Fast "Soft" Projections

In the previous section we introduced the learning apparatus. Our goal now is to derive and analyze an efficient algorithm for solving the label ranking problem. In addition to efficiency, we also require that the algorithm would be general and flexible so it can be used with *any* decomposition of the feedback according to $\mathbf{E}(\boldsymbol{\gamma})$. While the algorithm presented in this and the coming sections is indeed efficient and general, its derivation is rather complex. We therefore would like to present it in a bottom-up manner starting with a sub-problem which constitutes the main building block of the algorithm. In this sub-problem we assume that we have obtained a label-ranking function realized by the set $\mathbf{u}_1,\ldots,\mathbf{u}_k$ and the goal is to modify the ranking function so as to fit better a newly obtained example. To further simplify the derivation, we focus on the case where $\mathbf{E}(\boldsymbol{\gamma})$ contains a single complete bipartite graph whose set of edges are simply denoted by E. The end result is the following simplified constrained optimization problem,

$$\begin{aligned} \min_{\mathbf{w}_1,\ldots,\mathbf{w}_k,\xi} \quad & \frac{1}{2}\sum_{y=1}^{k}\|\mathbf{w}_y - \mathbf{u}_y\|^2 + C\xi \\ \text{s.t.} \quad & \forall (r,s) \in E,\ \ \mathbf{w}_r \cdot \mathbf{x} - \mathbf{w}_s \cdot \mathbf{x} \geq \gamma_r - \gamma_s - \xi \\ & \xi \geq 0 \ . \end{aligned} \tag{7}$$

Here $\mathbf{x} \in \mathcal{X}$ is a single instance and E is a set of edges which induces a complete bipartite graph.

The focus of this section is an efficient algorithm for solving Eq. (7). This optimization problem finds the set closest to $\{\mathbf{u}_1,\ldots,\mathbf{u}_k\}$ which approximately satisfies a system of linear constraints with a single slack (relaxation) variable ξ. Put another way, we can view the problem as the task of finding a relaxed projection of the set $\{\mathbf{u}_1,\ldots,\mathbf{u}_k\}$ onto the polyhedron defined by the set of linear constraints induced from E. We thus refer to this task as the soft projection. Our algorithmic solution, while being efficient, is rather detailed and its derivation consists of multiple complex steps. We therefore start with a high level overview of its derivation. We first derive a dual version

of the problem defined by Eq. (7). Each variable in the dual problem corresponds to an edge in E. Thus, the total number of dual variables can be as large as $k^2/4$. We then introduce a new and more compact optimization problem which has only k variables. We prove that the reduced problem nonetheless attains the same optimum as the original dual problem. This reduction is one of the two major steps in the derivation of an efficient soft projection procedure. We next show that the reduced problem can be decoupled into two simpler constrained optimization problems each of which corresponds to one layer in the bipartite graph induced by E. The two problems are tied by a single variable. We finally reach an efficient solution by showing that the optimal value of the coupling variable can be efficiently computed in $O(k \log(k))$ time. We recap our entire derivation by providing the pseudo-code of the resulting algorithm at the end of the section.

3.1 The Dual Problem

To start, we would like to note that the primal objective function is convex and all the primal constraints are linear. A necessary and sufficient condition for strong duality to hold in this case is that there exists a feasible solution to the primal problem (see for instance (Boyd and Vandenberghe, 2004)). A feasible solution can indeed obtained by simply setting $\mathbf{w}_y = \mathbf{0}$ for all y and defining $\xi = \max_{(r,s)\in E}(\gamma_r - \gamma_s)$. Therefore, strong duality holds and we can obtain a solution to the primal problem by finding the solution of its dual problem. To do so we first write the Lagrangian of the primal problem given in Eq. (7), which amounts to,

$$\begin{aligned}
\mathcal{L} &= \frac{1}{2}\sum_{y=1}^{k} \|\mathbf{w}_y - \mathbf{u}_y\|^2 + C\xi + \sum_{(r,s)\in E} \tau_{r,s}\left(\gamma_r - \gamma_s - \xi + \mathbf{w}_s \cdot \mathbf{x} - \mathbf{w}_r \cdot \mathbf{x}\right) - \zeta\xi \\
&= \frac{1}{2}\sum_{y=1}^{k} \|\mathbf{w}_y - \mathbf{u}_y\|^2 + \xi\left(C - \sum_{(r,s)\in E} \tau_{r,s} - \zeta\right) + \sum_{(r,s)\in E} \tau_{r,s}\left(\gamma_r - \mathbf{w}_r \cdot \mathbf{x} - \gamma_s + \mathbf{w}_s \cdot \mathbf{x}\right) \ ,
\end{aligned}$$

where $\tau_{r,s} \geq 0$ for all $(r,s) \in E$ and $\zeta \geq 0$. To derive the dual problem we now can use the strong duality. We eliminate the primal variables by minimizing the Lagrangian with respect to its primal variables. First, note that the minimum of the term $\xi(C - \sum_{(r,s)\in E}\tau_{r,s} - \zeta)$ with respect to ξ is zero whenever $C - \sum_{(r,s)\in E} \tau_{r,s} - \zeta = 0$. If however $C - \sum_{(r,s)\in E} \tau_{r,s} - \zeta \neq 0$ then this term can be made to approach $-\infty$. Since we need to maximize the dual we can rule out the latter case and pose the following constraint on the dual variables,

$$C - \sum_{(r,s)\in E} \tau_{r,s} - \zeta \ = \ 0 \ . \tag{8}$$

Next, recall our assumption that E induces a complete bipartite graph (V, E) (see also Eq. (2)). Therefore, there exists two sets A and B such that $A \cap B = \emptyset$, $V = A \cup B$, and $E = A \times B$. Using the definition of the sets A and B we can rewrite the last sum of the Lagrangian as,

$$\begin{aligned}
&\sum_{r\in A, s\in B} \tau_{r,s}\left(\gamma_r - \mathbf{w}_r \cdot \mathbf{x} - \gamma_s + \mathbf{w}_s \cdot \mathbf{x}\right) = \\
&\qquad \sum_{r\in A}\left(\gamma_r - \mathbf{w}_r \cdot \mathbf{x}\right)\sum_{s\in B} \tau_{r,s} \ - \ \sum_{s\in B}\left(\gamma_s - \mathbf{w}_s \cdot \mathbf{x}\right)\sum_{r\in A} \tau_{r,s} \ .
\end{aligned}$$

Eliminating the remaining primal variables $\mathbf{w}_1, \ldots, \mathbf{w}_k$ is done by differentiating the Lagrangian with respect to $\mathbf{w}_r$ for all $r \in [k]$ and setting the result to zero. For all $y \in A$, the above gives the set of constraints,

$$\nabla_{\mathbf{w}_y} \mathcal{L} \;=\; \mathbf{w}_y - \mathbf{u}_y - \left(\sum_{s \in B} \tau_{y,s} \right) \mathbf{x} \;=\; 0 \;. \tag{9}$$

Similarly, for $y \in B$ we get that,

$$\nabla_{\mathbf{w}_y} \mathcal{L} \;=\; \mathbf{w}_y - \mathbf{u}_y + \left(\sum_{r \in A} \tau_{r,y} \right) \mathbf{x} \;=\; 0 \;. \tag{10}$$

Finally, we would like to note that for any label $y \notin A \cup B$ we get that $\mathbf{w}_y - \mathbf{u}_y = 0$. Thus, we can omit all such labels from our derivation. Summing up, we get that,

$$\mathbf{w}_y \;=\; \begin{cases} \mathbf{u}_y + \left(\sum_{s \in B} \tau_{y,s} \right) \mathbf{x} & y \in A \\ \mathbf{u}_y - \left(\sum_{r \in A} \tau_{r,y} \right) \mathbf{x} & y \in B \\ \mathbf{u}_y & \text{otherwise} \end{cases} \;. \tag{11}$$

Plugging Eq. (11) and Eq. (8) into the Lagrangian and rearranging terms give the following dual objective function,

$$\begin{aligned} D(\boldsymbol{\tau}) \quad = \quad & -\frac{1}{2} \left\| \mathbf{x} \right\|^2 \sum_{y \in A} \left(\sum_{s \in B} \tau_{y,s} \right)^2 - \frac{1}{2} \left\| \mathbf{x} \right\|^2 \sum_{y \in B} \left(\sum_{r \in A} \tau_{r,y} \right)^2 \\ & + \sum_{y \in A} \left(\gamma_y - \mathbf{u}_y \cdot \mathbf{x} \right) \sum_{s \in B} \tau_{y,s} - \sum_{y \in B} \left(\gamma_y - \mathbf{u}_y \cdot \mathbf{x} \right) \sum_{r \in A} \tau_{r,y} \;. \end{aligned} \tag{12}$$

In summary, the resulting dual problem is,

$$\max_{\boldsymbol{\tau} \in \mathbb{R}_+^{|E|}} \; D(\boldsymbol{\tau}) \qquad \text{s.t.} \quad \sum_{(r,s) \in E} \tau_{r,s} \leq C \;. \tag{13}$$

3.2 Reparametrization of the Dual Problem

Each dual variable $\tau_{r,s}$ corresponds to an edge in E. Thus, the number of dual variables may be as large as $k^2/4$. However, the dual objective function depends only on sums of variables $\tau_{r,s}$. Furthermore, each primal vector $\mathbf{w}_y$ also depends on sums of dual variables (see Eq. (11)). We exploit these useful properties to introduce an equivalent optimization of a smaller size with only k variables. We do so by defining the following variables,

$$\forall y \in A, \;\; \alpha_y = \sum_{s \in B} \tau_{y,s} \qquad \text{and} \qquad \forall y \in B, \;\; \beta_y = \sum_{r \in A} \tau_{r,y} \;. \tag{14}$$

The primal variables $\mathbf{w}_y$ from Eq. (11) can be rewritten using α_y and β_y as follows,

$$\mathbf{w}_y \;=\; \begin{cases} \mathbf{u}_y + \alpha_y \mathbf{x} & y \in A \\ \mathbf{u}_y - \beta_y \mathbf{x} & y \in B \\ \mathbf{u}_y & \text{otherwise} \end{cases} \;. \tag{15}$$

Overloading our notation and using $D(\boldsymbol{\alpha}, \boldsymbol{\beta})$ to denote dual objective function in terms of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we can rewrite the dual objective of Eq. (12) as follows,

$$D(\boldsymbol{\alpha}, \boldsymbol{\beta}) \;= -\frac{1}{2}\|\mathbf{x}\|^2 \left(\sum_{y \in A} \alpha_y^2 + \sum_{y \in B} \beta_y^2 \right) + \sum_{y \in A} (\gamma_y - \mathbf{u}_y \cdot \mathbf{x})\, \alpha_y - \sum_{y \in B} (\gamma_y - \mathbf{u}_y \cdot \mathbf{x})\, \beta_y \;. \quad (16)$$

Note that the definition of α_y and β_y from Eq. (14) implies that α_y and β_y are non-negative. Furthermore, by construction we also get that,

$$\sum_{y \in A} \alpha_y \;=\; \sum_{y \in B} \beta_y \;=\; \sum_{(r,s) \in E} \tau_{r,s} \;\leq\; C \;. \quad (17)$$

In summary, we have obtained the following constrained optimization problem,

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}_+^{|A|},\, \boldsymbol{\beta} \in \mathbb{R}_+^{|B|}} D(\boldsymbol{\alpha}, \boldsymbol{\beta}) \quad \text{s.t.} \quad \sum_{y \in A} \alpha_y = \sum_{y \in B} \beta_y \leq C \;. \quad (18)$$

We refer to the above optimization problem as the *reduced* problem since it encompasses at most $k = |V|$ variables. In appendix A we show that the reduced problem and the original dual problem from Eq. (13) are equivalent. The end result is the following corollary.

Corollary 1 *Let $(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star)$ be the optimal solution of the reduced problem in Eq. (18). Define $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ as in Eq. (15). Then, $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ is the optimal solution of the soft projection problem defined by Eq. (7).*

We now move our focus to the derivation of an efficient algorithm for solving the reduced problem. To make our notation easy to follow, we define $p = |A|$ and $q = |B|$ and construct two vectors $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\nu} \in \mathbb{R}^q$ such that for each $a \in A$ there is an element $(\gamma_a - \mathbf{u}_a \cdot \mathbf{x})/\|\mathbf{x}\|^2$ in $\boldsymbol{\mu}$ and for each $b \in B$ there is an element $-(\gamma_b - \mathbf{u}_b \cdot \mathbf{x})/\|\mathbf{x}\|^2$ in $\boldsymbol{\nu}$. The reduced problem can now be rewritten as,

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}_+^p, \boldsymbol{\beta} \in \mathbb{R}_+^q} \quad & \frac{1}{2}\|\boldsymbol{\alpha} - \boldsymbol{\mu}\|^2 + \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{\nu}\|^2 \\ \text{s.t.} \quad & \sum_{i=1}^{p} \alpha_i \;=\; \sum_{j=1}^{q} \beta_j \;\leq\; C \;. \end{aligned} \quad (19)$$

3.3 Decoupling the reduced optimization problem

In the previous section we showed that the soft projection problem given by Eq. (7) is equivalent to the reduced optimization problem of Eq. (19). Note that the variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are tied together through a single equality constraint $\|\boldsymbol{\alpha}\|_1 = \|\boldsymbol{\beta}\|_1$. We represent this coupling of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by rewriting the optimization problem in Eq. (19) as,

$$\min_{z \in [0,C]} \quad g(z; \boldsymbol{\mu}) + g(z; \boldsymbol{\nu}) \;,$$

where

$$g(z; \boldsymbol{\mu}) \;=\; \min_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{\alpha} - \boldsymbol{\mu}\|^2 \quad \text{s.t.} \quad \sum_{i=1}^{p} \alpha_i \;=\; z \;,\;\; \alpha_i \geq 0 \;, \quad (20)$$

and similarly

$$g(z;\boldsymbol{\nu}) \;=\; \min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}-\boldsymbol{\nu}\|^2 \quad \text{s.t.} \quad \sum_{j=1}^{q}\beta_j \;=\; z \;,\;\; \beta_j \geq 0 \;. \tag{21}$$

The function $g(z;\cdot)$ takes the same functional form whether we use $\boldsymbol{\mu}$ or $\boldsymbol{\nu}$ as the second argument. We therefore describe our derivation in terms of $g(z;\boldsymbol{\mu})$. Clearly, the same derivation is also applicable to $g(z;\boldsymbol{\nu})$. The Lagrangian of $g(z;\boldsymbol{\mu})$ is,

$$\mathcal{L} \;=\; \frac{1}{2}\|\boldsymbol{\alpha}-\boldsymbol{\mu}\|^2 + \theta\left(\sum_{i=1}^{p}\alpha_i - z\right) - \boldsymbol{\zeta}\cdot\boldsymbol{\alpha} \;\;,$$

where $\theta \in \mathbb{R}$ is a Lagrange multiplier and $\boldsymbol{\zeta} \in \mathbb{R}^p_+$ is a vector of non-negative Lagrange multipliers. Differentiating with respect to α_i and comparing to zero gives the following KKT condition,

$$\frac{d\mathcal{L}}{d\alpha_i} \;=\; \alpha_i - \mu_i + \theta - \zeta_i \;=\; 0 \;.$$

The complementary slackness KKT condition implies that whenever $\alpha_i > 0$ we must have that $\zeta_i = 0$. Thus, if $\alpha_i > 0$ we get that,

$$\alpha_i \;=\; \mu_i - \theta + \zeta_i \;=\; \mu_i - \theta \;. \tag{22}$$

Since all the non-negative elements of the vector $\boldsymbol{\alpha}$ are tied via a single variable we would have ended with a much simpler problem had we known the indices of these elements. On a first sight, this task seems difficult as the number of potential subsets of $\boldsymbol{\alpha}$ is clearly exponential in the dimension of $\boldsymbol{\alpha}$. Fortunately, the particular form of the problem renders an efficient algorithm for identifying the non-zero elements of $\boldsymbol{\alpha}$. The following lemma is a key tool in deriving our procedure for identifying the non-zero elements.

Lemma 2 *Let $\boldsymbol{\alpha}$ be the optimal solution to the minimization problem in Eq. (20). Let s and j be two indices such that $\mu_s > \mu_j$. If $\alpha_s = 0$ then α_j must be zero as well.*

Proof Assume by contradiction that $\alpha_s = 0$ yet $\alpha_j > 0$. Let $\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^k$ be a vector whose elements are equal to the elements of $\boldsymbol{\alpha}$ except for $\tilde{\alpha}_s$ and $\tilde{\alpha}_j$ which are interchanged, that is, $\tilde{\alpha}_s = \alpha_j$, $\tilde{\alpha}_j = \alpha_s$, and for every other $r \notin \{s,j\}$ we have $\tilde{\alpha}_r = \alpha_r$. It is immediate to verify that the constraints of Eq. (20) still hold. In addition we have that,

$$\|\boldsymbol{\alpha}-\boldsymbol{\mu}\|^2 - \|\tilde{\boldsymbol{\alpha}}-\boldsymbol{\mu}\|^2 \;=\; \mu_s^2 + (\alpha_j-\mu_j)^2 - (\alpha_j-\mu_s)^2 - \mu_j^2 \;=\; 2\alpha_j(\mu_s-\mu_j) \;>\; 0 \;.$$

Therefore, we obtain that $\|\boldsymbol{\alpha}-\boldsymbol{\mu}\|^2 > \|\tilde{\boldsymbol{\alpha}}-\boldsymbol{\mu}\|^2$, which contradicts the fact that $\boldsymbol{\alpha}$ is the optimal solution. ■

Let I denote the set $\{i \in [p] : \alpha_i > 0\}$. The above lemma gives a simple characterization of the set I. Let us reorder the $\boldsymbol{\mu}$ such that $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_p$. Simply put, Lemma 2 implies that after the reordering, the set I is of the form $\{1,\ldots,\rho\}$ for some $1 \leq \rho \leq p$. Had we known ρ we could have simply use Eq. (22) and get that

$$\sum_{i=1}^{p}\alpha_i = \sum_{i=1}^{\rho}\alpha_i = \sum_{i=1}^{\rho}(\mu_i-\theta) = z \;\;\Rightarrow\;\; \theta = \frac{1}{\rho}\left(\sum_{i=1}^{\rho}\mu_i - z\right) \;\;.$$

In summary, given ρ we can summarize the optimal solution for $\boldsymbol{\alpha}$ as follows,

$$\alpha_i \;=\; \begin{cases} \mu_i - \frac{1}{\rho}\left(\sum_{i=1}^{\rho}\mu_i - z\right) & i \le \rho \\ 0 & i > \rho \end{cases} \quad . \tag{23}$$

We are left with the problem of finding the optimal value of ρ. We could simply enumerate all possible values of ρ in $[p]$, for each possible value compute $\boldsymbol{\alpha}$ as given by Eq. (23), and then choose the value for which the objective function ($\|\boldsymbol{\alpha} - \boldsymbol{\mu}\|^2$) is the smallest. While this procedure can be implemented quite efficiently, the following lemma provides an even simpler solution once we reorder the elements of $\boldsymbol{\mu}$ to be in a non-increasing order.

Lemma 3 *Let $\boldsymbol{\alpha}$ be the optimal solution to the minimization problem given in Eq. (20) and assume that $\mu_1 \ge \mu_2 \ge \ldots \ge \mu_p$. Then, the number of strictly positive elements in $\boldsymbol{\alpha}$ is,*

$$\rho(z, \boldsymbol{\mu}) \;=\; \max\left\{ j \in [p] \;:\; \mu_j - \frac{1}{j}\left(\sum_{r=1}^{j}\mu_r - z\right) > 0 \right\} \quad .$$

The proof of this technical lemma is deferred to the appendix.

Had we known the optimal value of z, i.e. the argument attaining the minimum of $g(z;\boldsymbol{\mu}) + g(z;\boldsymbol{\nu})$ we could have calculated the optimal dual variables $\boldsymbol{\alpha}^\star$ and $\boldsymbol{\beta}^\star$ by first finding $\rho(z,\boldsymbol{\mu})$ and $\rho(z,\boldsymbol{\nu})$ and then finding $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ using Eq. (23). This is a classical chicken-and-egg problem: we can easily calculate the optimal solution given some side information, however, obtaining the side information seems as difficult as finding the optimal solution. One option is to perform a search over an ϵ-net of values for z in $[0, C]$. For each candidate value for z from the ϵ-net we can find $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and then choose the value which attains the lowest objective value ($g(z;\boldsymbol{\mu}) + g(z;\boldsymbol{\nu})$). While this approach may be viable in many cases, it is still quite time consuming. To our rescue comes the fact that $g(z;\boldsymbol{\mu})$ and $g(z;\boldsymbol{\nu})$ entertain a very special structure. Rather than enumerating over all possible values of z we need to check at most $k+1$ possible values for z. To establish the last part of our efficient algorithm which performs this search for the optimal value of z we need the following theorem. The theorem is stated with $\boldsymbol{\mu}$ but, clearly, it also holds for $\boldsymbol{\nu}$.

Theorem 4 *Let $g(z;\boldsymbol{\mu})$ be as defined in Eq. (20). For each $i \in [p]$, define*

$$z_i = \sum_{r=1}^{i}\mu_r - i\mu_i \quad .$$

Then, for each $z \in [z_i, z_{i+1}]$ the function $g(z;\boldsymbol{\mu})$ is equivalent to the following quadratic function,

$$g(z;\boldsymbol{\mu}) \;=\; \frac{1}{i}\left(\sum_{r=1}^{i}\mu_r - z\right)^2 + \sum_{r=i+1}^{p}\mu_r^2 \quad .$$

Moreover, g is continuous, continuously differentiable, and convex in $[0, C]$.

The proof of this theorem is also deferred to the appendix. The good news that the theorem carries is that $g(z;\boldsymbol{\mu})$ and $g(z;\boldsymbol{\nu})$ are convex and therefore their sum is also convex. Furthermore, the function $g(z;\cdot)$ is piecewise quadratic and the points where it changes from one quadratic function to another are simple to compute. We refer to these points as knots. In the next sub-section we exploit the properties of the function g to devise an efficient procedure for finding the optimal value of z and from there the road to the optimal dual variables is clear and simple.

INPUT: instance $\mathbf{x} \in \mathcal{X}$; target ranking $\boldsymbol{\gamma}$; sets A, B

current prototypes $\mathbf{u}^1, \ldots, \mathbf{u}^k$; regularization parameter C

MARGINS:

$$\boldsymbol{\mu} = \text{sort}\left\{(\gamma_a - \mathbf{u}^a \cdot \mathbf{x})/\|\mathbf{x}\|^2 \mid a \in A\right\}$$
$$\boldsymbol{\nu} = \text{sort}\left\{(\mathbf{u}^b \cdot \mathbf{x} - \gamma_b)/\|\mathbf{x}\|^2 \mid b \in B\right\}$$

KNOTS:

$$\forall i \in [p] \; : \; z_i = \textstyle\sum_{r=1}^{i} \mu_r - i\mu_i \qquad \forall j \in [q] \; : \; \tilde{z}_j = \sum_{s=1}^{j} \nu_s - j\nu_j$$
$$\mathcal{Q} = \{z_i : z_i < C\} \cup \{\tilde{z}_j : \tilde{z}_j < C\} \cup \{C\}$$

INTERVALS:

$$\forall z \in \mathcal{Q} : \quad R(z) = |\{z_i : z_i \leq z\}| \quad ; \quad S(z) = |\{\tilde{z}_j : \tilde{z}_j \leq z\}|$$
$$\forall z \in \mathcal{Q} : \quad N(z) = \min\{z' \in \mathcal{Q} : z' > z\} \cup \{C\}$$

LOCAL MIN:

$$O(z) \; = \; \left(S(z) \sum_{r=1}^{R(z)} \mu_r + R(z) \sum_{r=1}^{S(z)} \nu_r \right) / \left(R(z) + S(z)\right)$$

GLOBAL MIN:

If $(\exists z \in \mathcal{Q}$ s.t. $O(z) \in [z, N(z)])$ **Then**

$z^\star = O(z)$; $i^\star = R(z)$; $j^\star = S(z)$

Else If $(\mu_1 + \nu_1 \leq 0)$

$z^\star = 0$; $i^\star = 1$; $j^\star = 1$

Else

$z^\star = C$; $i^\star = R(C)$; $j^\star = S(C)$

DUAL'S AUXILIARIES:

$$\theta_\alpha = \frac{1}{i^\star}\left(\sum_{r=1}^{i^\star} \mu_r - z^\star\right) \; ; \; \theta_\beta = \frac{1}{j^\star}\left(\sum_{r=1}^{j^\star} \nu_r - z^\star\right)$$

OUTPUT:

$$\forall a \in A \; : \; \alpha_a \; = \; \left(\tfrac{\gamma_a - u_a \cdot \mathbf{x}}{\|\mathbf{x}\|^2} - \theta_\alpha\right)_+ \text{ and } \; \mathbf{w}_a \; = \; \mathbf{u}_a + \alpha_a \, \mathbf{x}$$
$$\forall b \in B \; : \; \beta_b \; = \; \left(\tfrac{u_b \cdot \mathbf{x} - \gamma_b}{\|\mathbf{x}\|^2} - \theta_\beta\right)_+ \text{ and } \; \mathbf{w}_b \; = \; \mathbf{u}_b - \beta_b \, \mathbf{x}$$

Figure 3: Pseudo-code of the soft-projection onto polyhedra (SOPOPO) algorithm.

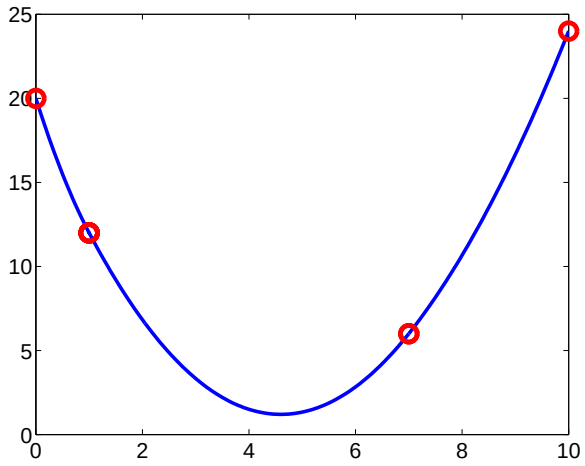

Figure 4: An illustration of the function $g(z;\boldsymbol{\mu}) + g(z;\boldsymbol{\nu})$. The vectors $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are constructed from, $\boldsymbol{\gamma} = (1, 2, 3, 4, 5, 6)$, $\mathbf{u} \cdot \mathbf{x} = (2, 3, 5, 1, 6, 4)$, $A = \{4, 5, 6\}$, and $B = \{1, 2, 3\}$.

3.4 Putting it all together

Due to the strict convexity of $g(z;\boldsymbol{\mu}) + g(z;\boldsymbol{\nu})$ its minimum is unique and well defined. Therefore, it suffices to search for a seemingly *local* minimum over all the sub-intervals in which the objective function is equivalent to a quadratic function. If such a local minimum point is found it is guaranteed to be the global minimum. Once we have the value of z which constitutes the global minimum we can decouple the optimization problems for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and quickly find the optimal solution. There is though one last small obstacle: the objective function is the sum of two piecewise quadratic functions. We therefore need to efficiently go over the *union* of the knots derived from $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. We now summarize the full algorithm for finding the optimum of the dual variables and wrap up with its pseudo-code.

Given $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ we find the sets of knots for each vector, take the union of the two sets, and sort the set in an ascending order. Based on the theorems above, it follows immediately that each interval between two consecutive knots in the union is also quadratic. Since $g(z;\boldsymbol{\mu}) + g(z;\boldsymbol{\nu})$ is convex, the objective function in each interval can be characterized as falling into one of two cases. Namely, the objective function is either monotone (increasing or decreasing) or it attains its unique global minimum inside the interval. In the latter case the objective function clearly decreases, reaches the optimum where its derivative is zero, and then increases. See also Fig. 4 for an illustration. If the objective function is monotone in all of the intervals then the minimum is obtained at one of the boundary points $z = 0$ or $z = C$. Otherwise, we simply need to identify the interval bracketing the global minimum and then find the optimal value of z by finding the minimizer of the quadratic function associated with the interval. For instance, in Fig. 4 the minimum is attained just below 5 at the interval defined by the second and third knots. If C is, say, 10 then the optimal value for z coincides with the minimum below 5. If however C lies to the left of the minimum, say at 3, then the optimum of z is at 3. We now formally recap the entire procedure.

We utilize the following notation. For each $i \in [p]$, define the knots derived from $\boldsymbol{\mu}$

$$z_i = \sum_{r=1}^{i} \mu_r - i\mu_i \ ,$$

and similarly, for each $j \in [q]$ define

$$\tilde{z}_j = \sum_{r=1}^{j} \nu_r - j\nu_j \ .$$

From Lemma 4 we know that $g(z;\boldsymbol{\mu})$ is quadratic in each segment $[z_i, z_{i+1})$ and $g(z;\boldsymbol{\nu})$ is quadratic in each segment $[\tilde{z}_j, \tilde{z}_{j+1})$. Therefore, as already argued above, the function $g(z;\boldsymbol{\mu}) + g(z;\boldsymbol{\nu})$ is also piecewise quadratic in $[0, C]$ and its knots are the points in the set,

$$\mathcal{Q} = \{z_i : z_i < C\} \cup \{\tilde{z}_j : \tilde{z}_j < C\} \cup \{C\} \ .$$

For each knot $z \in \mathcal{Q}$, we denote by $N(z)$ its consecutive knot in $\mathcal{Q}$, that is,

$$N(z) = \min \ \left(\{z' \in \mathcal{Q} : z' > z\} \cup \{C\}\right) \ .$$

We also need to know for each knot how many knots precede it. Given a knot z we define

$$R(z) = |\{z_i : z_i \leq z\}| \ \text{ and } \ S(z) = |\{\tilde{z}_i : \tilde{z}_i \leq z\}| \ .$$

Using the newly introduced notation we can find for a given value z its bracketing interval, $z \in [z', N(z')]$. From Thm. 4 we get that the value of the dual objective function at z is,

$$g(z;\boldsymbol{\mu}) + g(z;\boldsymbol{\nu}) \ = \\ \frac{1}{R(z')}\left(\sum_{r=1}^{R(z')} \mu_r - z\right)^2 + \sum_{r=R(z')+1}^{p} \mu_r^2 + \frac{1}{S(z')}\left(\sum_{r=1}^{S(z')} \nu_r - z\right)^2 + \sum_{r=S(z')+1}^{p} \nu_r^2 \ .$$

The unique minimum of the quadratic function above is attained at the point

$$O(z') = \left(S(z') \sum_{i=1}^{R(z')} \mu_i + R(z') \sum_{i=1}^{S(z')} \nu_i\right) / \left(R(z') + S(z')\right) \ .$$

Therefore, if $O(z') \in [z', N(z')]$, then the global minimum of the dual objective function is attained at $O(z')$. Otherwise, if no such interval exists, the optimum is either at $z = 0$ or at $z = C$. The minimum is achieved at $z = 0$ iff the derivative of the objective function at $z = 0$ is non-negative, namely, $-\mu_1 - \nu_1 \geq 0$. In this case, the optimal solution is $\boldsymbol{\alpha} = \mathbf{0}$ and $\boldsymbol{\beta} = \mathbf{0}$ which implies that $\mathbf{w}_r = \mathbf{u}_r$ for all r. If on the other hand $-\mu_1 - \nu_1 < 0$ then the optimum is attained at $z = C$. The skeleton of the pseudo-code for the fast projection algorithm is given in Fig. 3. The most expensive operation performed by the algorithm is the sorting of $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. Since the sum of the dimensions of these vectors is k the time complexity of the algorithm is $\Theta(k \log(k))$.

4. From a single projection to multiple projections

We now describe the algorithm for solving the original batch problem defined by Eq. (6) using the SOPOPO algorithm as its core. We would first like to note that the general batch problem can also be viewed as a soft projection problem. We can cast the batch problem as finding the set of vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ which is closest to k zero vectors $\{\mathbf{0}, \ldots, \mathbf{0}\}$ while approximately satisfying a set of systems of linear constraints where each system is associated with an independent relaxation variable. Put another way, we can view the full batch optimization problem as the task of finding a relaxed projection of the set $\{\mathbf{0}, \ldots, \mathbf{0}\}$ onto multiple polyhedra each of which is defined via a set of linear constraints induced by a single sub-graph $E_j \in \mathbf{E}(\boldsymbol{\gamma}^i)$. We thus refer to this task as the soft-projection onto multiple polyhedra. We devise an iterative algorithm which solves the batch problem by successively calling to the SOPOPO algorithm from Fig. 3. We describe and analyze the algorithm for a slightly more general constrained optimization which results in a simplified notation. We start with the presentation of our original formulation as an instance of the generalized problem.

To convert the problem in Eq. (6) to a more general form, we assume without loss of generality that $|\mathbf{E}(\boldsymbol{\gamma}^i)| = 1$ for all $i \in [m]$. We refer to the single set in $\mathbf{E}(\boldsymbol{\gamma}^i)$ as E^i. This assumption does not pose a limitation since in the case of multiple decompositions, $\mathbf{E}(\boldsymbol{\gamma}^i) = \{E_1, \ldots, E_d\}$, we can replace the ith example with d pseudo-examples: $\{(\mathbf{x}^i, E_1), \ldots, (\mathbf{x}^i, E_d)\}$. Using this assumption, we can rewrite the optimization problem of Eq. (6) as follows,

$$\begin{aligned} \min_{\mathbf{w}_1,\ldots,\mathbf{w}_k,\boldsymbol{\xi}} \quad & \frac{1}{2}\sum_{r=1}^{k}\|\mathbf{w}_r\|^2 \;+\; \sum_{i=1}^{m} C_i\,\xi^i \\ \text{s.t.} \quad & \forall\, i \in [m],\, \forall\,(r,s) \in E^i, \quad \mathbf{w}_r \cdot \mathbf{x}^i - \mathbf{w}_s \cdot \mathbf{x}^i \geq \gamma^i_r - \gamma^i_s - \xi^i \\ & \forall\, i, \;\; \xi^i \geq 0 \;\; , \end{aligned} \tag{24}$$

where $C_i = C\sigma^i$ is the weight of the ith slack variable. To further simplify Eq. (24), we use $\bar{\mathbf{w}} \in \mathbb{R}^{nk}$ to denote the concatenation of the vectors $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$. In addition, we associate an index, denoted j, with each $(r,s) \in E^i$ and define $\mathbf{a}^{i,j} \in \mathbb{R}^{nk}$ to be the vector,

$$\mathbf{a}^{i,j} = (\; \underbrace{\mathbf{0}}_{\text{1st block}} \;,\; \ldots \;,\; \mathbf{0} \;,\; \underbrace{\mathbf{x}^i}_{r\text{th block}} \;,\; \mathbf{0} \;,\; \;,\; \ldots \;,\; \mathbf{0} \;,\; \underbrace{-\mathbf{x}^i}_{s\text{th block}} \;,\; \mathbf{0} \;,\; \ldots \;,\; \underbrace{\mathbf{0}}_{k\text{th block}} \;) \;. \tag{25}$$

We also define $b^{i,j} = \gamma^i_r - \gamma^i_s$. Finally, we define $k_i = |E^i|$. Using the newly introduced notation we can rewrite Eq. (24) as follows,

$$\begin{aligned} \min_{\bar{\mathbf{w}},\boldsymbol{\xi}} \quad & \frac{1}{2}\|\bar{\mathbf{w}}\|^2 \;+\; \sum_{i=1}^{m} C_i\,\xi^i \\ \text{s.t.} \quad & \forall i \in [m],\, \forall j \in [k_i], \;\; \bar{\mathbf{w}} \cdot \mathbf{a}^{i,j} \geq b^{i,j} - \xi^i \\ & \xi^i \geq 0 \;\; . \end{aligned} \tag{26}$$

Our goal is to derive an iterative algorithm for solving Eq. (26) based on a procedure for solving a single soft-projection which takes the form,

$$\begin{aligned} \min_{\bar{\mathbf{w}},\xi^i} \quad & \frac{1}{2}\|\bar{\mathbf{w}}-\mathbf{u}\|^2 + C_i\,\xi^i \\ \text{s.t.} \quad & \forall j \in [k_i],\ \ \bar{\mathbf{w}}\cdot\mathbf{a}^{i,j} \geq b^{i,j} - \xi^i \\ & \xi^i \geq 0 \ . \end{aligned} \tag{27}$$

By construction, an algorithm for solving the more general problem defined in Eq. (26) would also solve the more specific problem defined by Eq. (6).

The rest of the section is organized as follows. We first derive the dual of the problem given in Eq. (26). We then describe an iterative algorithm which on each iteration performs a single soft-projection and present a pseudo-code of the iterative algorithm tailored for the specific label-ranking problem of Eq. (6). Finally, we analyze the convergence of the suggested iterative algorithm.

4.1 The dual problem

First, note that the primal objective function of the general problem is convex and all the primal constraints are linear. Therefore, using the same arguments as in Sec. 3.1 it is simple to show that strong duality holds and a solution to the primal problem can be obtained from the solution of its dual problem. To derive the dual problem, we first write the Lagrangian,

$$\mathcal{L} = \frac{1}{2}\|\bar{\mathbf{w}}\|^2 + \sum_{i=1}^{m} C_i\,\xi^i + \sum_{i=1}^{m}\sum_{j=1}^{k_i}\lambda_{i,j}\left(b^{i,j} - \xi^i - \bar{\mathbf{w}}\cdot\mathbf{a}^{i,j}\right) - \sum_{i=1}^{m}\zeta_i\xi^i \ ,$$

where $\lambda_{i,j}$ and ζ_i are non-negative Lagrange multipliers. Taking the derivative of $\mathcal{L}$ with respect to $\bar{\mathbf{w}}$ and comparing it to zero gives,

$$\bar{\mathbf{w}} \ = \ \sum_{i=1}^{m}\sum_{j=1}^{k_i}\lambda_{i,j}\,\mathbf{a}^{i,j} \ . \tag{28}$$

As in the derivation of the dual objective function for a single soft projection, we get that the following must hold at the optimum,

$$\forall i \in [m], \quad \sum_{j=1}^{k_i}\lambda_{i,j} - C_i - \zeta_i = 0 \ . \tag{29}$$

Since $\lambda_{i,j}$ and ζ_i are non-negative Lagrange multipliers we get that the set of feasible solutions of the dual problem is,

$$S \ = \ \left\{ \boldsymbol{\lambda} \ \middle| \ \forall i,\ \sum_{j=1}^{k_i}\lambda_{i,j} \ \leq \ C_i \ \text{ and } \ \forall i,j,\ \lambda_{i,j} \geq 0 \right\} \ .$$

Using Eq. (28) and Eq. (29) to further rewrite the Lagrangian gives the dual objective function,

$$D(\boldsymbol{\lambda}) \ = \ -\frac{1}{2}\left\|\sum_{i=1}^{m}\sum_{j=1}^{k_i}\lambda_{i,j}\,\mathbf{a}^{i,j}\right\|^2 + \sum_{i=1}^{m}\sum_{j=1}^{k_i}\lambda_{i,j}\,b^{i,j} \ .$$

INPUT: training set $\{(\mathbf{x}^i, \boldsymbol{\gamma}^i)\}_{i=1}^m$; decomposition function $\mathbf{E}(\boldsymbol{\gamma})$; regularization parameter C

INITIALIZE:

$\forall i \in [m],\ A_j \times B_j \in \mathbf{E}(\boldsymbol{\gamma}^i),\ (a,b) \in A_j \times B_j,$ set $\alpha_a^{i,j} = 0,\ \beta_b^{i,j} = 0$

$\forall r \in [k]$, set $\mathbf{w}_r = \mathbf{0}$

LOOP:

Choose a sub-graph $i \in [m]$, $A_j \times B_j \in \mathbf{E}(\boldsymbol{\gamma}^i)$

UPDATE:

$$\forall a \in A_j :\ \mathbf{u}_a \ = \ \mathbf{w}_a - \alpha_a^{i,j}\mathbf{x}_i \quad \forall b \in B_j :\ \mathbf{u}_b \ = \ \mathbf{w}_b + \beta_b^{i,j}\mathbf{x}_i$$

SOLVE:

$$(\boldsymbol{\alpha}^{i,j}, \boldsymbol{\beta}^{i,j}, \{\mathbf{w}_r\}) = \text{SOPOPO}(\{\mathbf{u}_r\}, \mathbf{x}^i, \boldsymbol{\gamma}^i, A_j, B_j, C\,\sigma_j^i)$$

OUTPUT: The final vectors $\{\mathbf{w}_r\}_{r=1}^k$

Figure 5: The procedure for solving the preference graphs problem via soft-projections.

The dual of the problem defined in Eq. (26) is therefore,

$$\max_{\boldsymbol{\lambda} \in S} \ D(\boldsymbol{\lambda}) \ . \tag{30}$$

4.2 An iterative procedure

We are now ready to describe our iterative algorithm. We would like to stress again that the methodology and analysis presented here have been suggested by several authors. Our procedure is a slight generalization of row action methods (Censor and Zenios, 1997) which is often referred to as decomposition methods (see also Lin (2002), Mangasarian and Musicant (1999), Platt (1998)). The iterative procedure works in rounds and operates on the dual form of the objective function. We show though that each round can be realized as a soft-projection operation. Let $\boldsymbol{\lambda}^t$ denote the vector of dual variables before the tth iteration of the iterative algorithm. Initially, we set $\boldsymbol{\lambda}^1 = \mathbf{0}$, which constitutes a trivial feasible solution to Eq. (30). On the tth iteration of the algorithm, we choose a single example whose index is denoted r and update its dual variables. We freeze the rest of the dual variables at their current value. We cast the tth iteration as the following constrained optimization problem,

$$\boldsymbol{\lambda}^{t+1} \ = \ \operatorname*{argmax}_{\boldsymbol{\lambda} \in S} D(\boldsymbol{\lambda}) \quad \text{s.t. } \forall i \neq r,\ \forall j \in [k_i],\ \lambda_{i,j} = \lambda_{i,j}^t \ . \tag{31}$$

Note that $\boldsymbol{\lambda}^{t+1}$ is essentially the same as $\boldsymbol{\lambda}^t$ except for the variables corresponding to the rth example, namely, $\{\lambda_{r,j} \,|\, j \in [k_r]\}$. In order to explicitly write the objective function conveyed by

Eq. (31) let us introduce the following notation,

$$\mathbf{u} \;=\; \sum_{i\neq r}\sum_{j=1}^{k_i} \lambda_{i,j}^{t}\mathbf{a}^{i,j} \;\; . \tag{32}$$

The vector $\mathbf{u}$ is equal to the current estimate of $\bar{\mathbf{w}}$ excluding the contribution of the rth set of dual variables. With $\mathbf{u}$ on hand, we can rewrite the objective function of Eq. (31) as follows,

$$\begin{aligned} &-\frac{1}{2}\left\|\sum_{j=1}^{k_r}\lambda_{r,j}\mathbf{a}^{r,j}\right\|^2 \;-\; \left(\sum_{j=1}^{k_r}\lambda_{r,j}\mathbf{a}^{r,j}\right)\cdot\mathbf{u} - \frac{1}{2}\left\|\mathbf{u}\right\|^2 + \sum_{j=1}^{k_r}\lambda_{r,j}b^{r,j} + \sum_{i\neq r}\sum_{j=1}^{k_i}\lambda_{i,j}^{t}b^{i,j} \\ &\quad = -\frac{1}{2}\left\|\sum_{j=1}^{k_r}\lambda_{r,j}\mathbf{a}^{r,j}\right\|^2 + \sum_{j=1}^{k_r}\lambda_{r,j}\left(b^{r,j} - \mathbf{u}\cdot\mathbf{a}^{r,j}\right) + \Gamma \;\; , \end{aligned} \tag{33}$$

where Γ is a constant that does not depend on the variables in $\{\lambda_{r,j}\,|\,j\in[k_r]\}$. In addition the set of variables which are not fixed must reside in S, therefore,

$$\sum_{j=1}^{k_i}\lambda_{r,j} \;\leq\; C_r \;\text{ and }\; \forall j,\; \lambda_{r,j}\geq 0 \;\; . \tag{34}$$

The fortunate circumstances are that the optimization problem defined by Eq. (33) subject to the constraints given in Eq. (34) can be rephrased as a soft-projection problem. Concretely, let us define the following soft-projection problem,

$$\begin{aligned} \min_{\bar{\mathbf{w}},\xi^r} \quad & \frac{1}{2}\|\bar{\mathbf{w}}-\mathbf{u}\|^2 \;+\; C_r\,\xi^r \\ \text{s.t. } & \forall j\in[k_r],\;\; \bar{\mathbf{w}}\cdot\mathbf{a}^{r,j} \geq b^{r,j} - \xi^r \\ & \xi^r \geq 0 \;\; . \end{aligned} \tag{35}$$

The value of $\lambda_{r,j}^{t+1}$ is obtained from the optimal value of the dual problem of Eq. (35) as we now show. The Lagrangian of Eq. (35) is

$$\mathcal{L} \;=\; \frac{1}{2}\|\bar{\mathbf{w}}-\mathbf{u}\|^2 \;+\; C_r\,\xi^r + \sum_{j=1}^{k_r}\lambda_{r,j}\left(b^{r,j} - \xi^r - \bar{\mathbf{w}}\cdot\mathbf{a}^{r,j}\right) - \zeta_r\xi^r \;\; .$$

Differentiating with respect to $\bar{\mathbf{w}}$ and comparing to zero give,

$$\bar{\mathbf{w}} \;=\; \mathbf{u} + \sum_{j=1}^{k_r}\;\lambda_{r,j}\;\mathbf{a}^{r,j} \;\; .$$

As in the previous derivations of the dual objective functions we also get that,

$$C_r - \zeta_r - \sum_{j=1}^{k_r}\lambda_{r,j} \;=\; 0 \;\; ,$$

and thus the Lagrange multipliers must satisfy,

$$\sum_{j=1}^{k_r} \lambda_{r,j} \;\leq\; C_r \;\;.$$

Therefore, the dual problem of Eq. (35) becomes,

$$\begin{aligned} \max_{\boldsymbol{\lambda}_{r,\cdot}} \quad & -\frac{1}{2}\left\|\sum_{j=1}^{k_r} \lambda_{r,j}\,\mathbf{a}^{r,j}\right\|^2 + \sum_{j=1}^{k_r} \lambda_{r,j}\,(b^{r,j} - \mathbf{u}\cdot\mathbf{a}^{r,j}) \quad \text{s.t.} \\ & \sum_{j=1}^{k_r} \lambda_{r,j} \leq C_r \;\text{ and }\; \forall j,\; \lambda_{r,j} \geq 0 \;\;, \end{aligned} \tag{36}$$

which is *identical* to the problem defined by Eq. (33) subject to the constraints given by Eq. (34).

In summary, our algorithm works by updating one set of dual variables on each round while fixing the rest of the variables to their current values. Finding the optimal value of the unrestricted variables is achieved by defining an instantaneous soft-projection problem. The instantaneous soft-projection problem is readily solved using the machinery developed in the previous section. The pseudo-code of this iterative procedure is given in Fig. 5. It is therefore left to reason about the formal properties of the iterative procedure. From the definition of the update from Eq. (31) we clearly get that on each round we are guaranteed to increase the dual objective function unless we are already at the optimum. In the next subsection we show that this iterative paradigm converges to the global optimum of the dual objective function.

To conclude this section, we would like to note that a prediction of our label-ranking function is solely based on inner products between vectors from $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and an instance $\mathbf{x}$. In addition, as we have shown in the previous section, the solution of each soft projection takes the form $\mathbf{w}_a = \mathbf{u}_a + \alpha_a \mathbf{x}^i$ and $\mathbf{w}_b = \mathbf{u}_b - \beta_b \mathbf{x}^i$. Since we initially set all vectors to be the zero vector, we get that at each step of the algorithm all the vectors can be expressed as linear combinations of the instances. Thus, as in the case of support vector machines for classification problems, we can replace the inner product operation with any Mercer kernel (Vapnik, 1998).

4.3 Analysis of Convergence

To analyze the convergence of the iterative procedure we need to introduce a few more definitions. We denote by D^t the value of the dual objective function *before* the tth iteration and by $\Delta_t = D^{t+1} - D^t$ the increase in the dual on the tth iteration. We also denote by $\Delta^i(\boldsymbol{\lambda})$ the potential increase we have gained had we chosen the ith example for updating $\boldsymbol{\lambda}$. We assume that on each iteration of the algorithm, we choose an example, whose index is r, which attains the maximal increase in the dual, therefore $\Delta^r(\boldsymbol{\lambda}) = \max_i \Delta^i(\boldsymbol{\lambda}^t)$. Last, let $D^\star$ and $\boldsymbol{\lambda}^\star$ denote the optimal value and argument of the dual objective function. Our algorithm maximizes the dual objective on each iteration subject to the constraint that for all $i \neq r$ and $j \in [k_i]$, the variables $\lambda_{i,j}$ are kept intact. Therefore, the sequence $D^1, D^2, \ldots$ is monotonically non-decreasing.

To prove convergence we need the following lemma which says that if the algorithm is at sub-optimal solution then it will keep increasing the dual objective on the subsequent iteration.

Lemma 5 *Let $\boldsymbol{\lambda}$ be a suboptimal solution, $D(\boldsymbol{\lambda}) < D^\star$. Then there exists an example r for which $\Delta^r(\boldsymbol{\lambda}) > 0$.*

Proof Assume by contradiction that for all i, $\Delta^i(\boldsymbol{\lambda}) = 0$ and yet $D(\boldsymbol{\lambda}) < D^\star$. In this case we clearly have that $\boldsymbol{\lambda} \neq \boldsymbol{\lambda}^\star$. Let $\mathbf{v} = \boldsymbol{\lambda}^\star - \boldsymbol{\lambda}$ denote the difference between the optimal solution and the current solution and denote $h(\theta) = D(\boldsymbol{\lambda} + \theta\mathbf{v})$ the value of the dual obtained by moving along the direction $\mathbf{v}$ from $\boldsymbol{\lambda}$. Since $D(\boldsymbol{\lambda})$ is concave then so is h. Therefore, the line tangent to h at 0 resides above h at all points but $\theta = 0$. We thus get that, $h(0) + h'(0)\theta \geq h(\theta)$ and in particular for $\theta = 1$ we obtain,

$$h'(0) \geq h(1) - h(0) = D(\boldsymbol{\lambda}^\star) - D(\boldsymbol{\lambda}) > 0 \ .$$

Let ∇D denote the gradient of the dual objective at $\boldsymbol{\lambda}$. Since $h'(0) = \nabla D \cdot \mathbf{v}$ we get that,

$$\nabla D \cdot \mathbf{v} \ > \ 0 \ . \tag{37}$$

We now rewrite $\mathbf{v}$ as the sum of vectors,

$$\mathbf{v} = \sum_{i=1}^{m} \mathbf{z}^i \quad \text{where} \ \ z^i_{r,j} = \begin{cases} v_{r,j} & r = i \\ 0 & r \neq i \end{cases} \ .$$

In words, we rewrite $\mathbf{v}$ as the sum of vectors each of which corresponds to the dual variables appearing in a single soft-projection problem induced by the ith example. From the definition of $\mathbf{z}^i$ together with the form of the dual constraints we get that the vector $\boldsymbol{\lambda} + \mathbf{z}^i$ is also a feasible solution for the dual problem. Using the assumption that for all i, $\Delta^i(\boldsymbol{\lambda}) = 0$, we get that for each $\theta \in [0, 1]$, $D(\boldsymbol{\lambda}) \geq D(\boldsymbol{\lambda} + \theta\mathbf{z}^i)$. Analogously to h we define the scalar function $h_i(\theta) = D(\boldsymbol{\lambda} + \theta\mathbf{z}^i)$. Since h_i is derived from the dual problem by constraining the dual variables to reside on the line $\boldsymbol{\lambda} + \theta\mathbf{z}^i$, then as the function D, h_i is also continuously differentiable. The fact that $h_i(0) \geq h_i(\theta)$ for all $\theta \in [0, 1]$ now implies that $h'_i(0) \leq 0$. Furthermore, $\nabla D \cdot \mathbf{z}^i = h'_i(0) \leq 0$ for all i which gives,

$$\nabla D \cdot \mathbf{v} \ = \ \nabla D \cdot \sum_{i=1}^{m} \mathbf{z}^i \ = \ \sum_{i=1}^{m} \nabla D \cdot \mathbf{z}^i \ \leq \ 0 \ ,$$

which contradicts Eq. (37). ■

Equipped with the above lemma we are now ready to prove that the iterative algorithm converges to an optimal solution.

Theorem 6 *Let D^t denote the value of the dual objective after the t'th iteration of the algorithm defined in Eq. (31). Denote by $D^\star$ the optimum of the problem given in Eq. (30). Then, the sequence $D^1, D^2, \ldots, D^t, \ldots$ converges to $D^\star$.*

Proof Recall that the primal problem has a trivial feasible solution which is attained by setting $\bar{\mathbf{w}} = \mathbf{0}$ and $\xi^i = \max_j b^{i,j}$. For this solution the value of the primal problem is finite. Since the value of the dual problem cannot exceed the value of the primal problem we get that $D^\star < \infty$. Therefore, the sequence of dual objective values is a monotonic, non-decreasing, and upper bounded sequence, $D^1 \leq D^2 \leq \ldots \leq D^t \leq \ldots \leq D^\star < \infty$. Thus, this sequence converges to a limit which we denote by D'. It is left to show that $D' = D^\star$. Assume by contradiction that $D^\star - D' = \epsilon > 0$. The set of feasible dual solutions, S, is a compact set. Let $\Delta' : S \to \mathbb{R}$ be the average increase of the dual over all possible choices for an example to use for updating $\boldsymbol{\lambda}$,

$$\Delta'(\boldsymbol{\lambda}) = \frac{1}{m} \sum_i \Delta^i(\boldsymbol{\lambda}) \ .$$

On each iteration we have by construction that $\Delta_t \geq \Delta'(\boldsymbol{\lambda}^t)$. Define $A = \{\boldsymbol{\lambda} : D(\boldsymbol{\lambda}) > D^\star - \epsilon/2\}$. From the concavity of D we get that the set $S \setminus A$ is a compact set. Since Δ' is a continuous function it attains a minimum value over $S \setminus A$. Denote this minimum value by κ and let $\tilde{\boldsymbol{\lambda}}$ be the point which attains this minimum. From Lemma 5 we know that $\kappa > 0$ since otherwise $D(\tilde{\boldsymbol{\lambda}})$ would have equal to $D^\star$ which in turn contradicts the fact that $\tilde{\boldsymbol{\lambda}} \notin A$. Since for all t we know that $D^t \leq D' = D^\star - \epsilon$ we conclude that $\boldsymbol{\lambda}^t \in S \setminus A$. This fact implies that for all t ,

$$\Delta_t \geq \Delta'(\boldsymbol{\lambda}^t) \geq \Delta'(\tilde{\lambda}) = \kappa \ .$$

The above lower bound on the increase in the dual implies that the sequence $D^1, D^2, D^3, \ldots$ diverges to infinity and thus $D' = \infty$ which is in contradiction to the fact that $D' = D^\star - \epsilon < \infty$. ■

5. Experiments

In this section we compare the SOPOPO algorithm from Fig. 3 and our iterative procedure for soft-projection onto multiple polyhedra from Fig. 5 to a commercial interior point method called LOQO (Vanderbei, 1999).

Our first set of experiments focuses on assessing the efficiency of SOPOPO for soft-projection onto a *single* polyhedron. In this set of experiments, the data was generated as follows. First, we chose the number of classes $k = |\mathcal{Y}|$ and defined E to be the set $A \times B$ with $A = [k/2]$ and $B = [k] \setminus [k/2]$. We set the value of γ_r to be one for $r \in A$ and otherwise it was set to zero. We then sampled an instance $\mathbf{x}$ and a set of vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ from a 100-dimensional Normal distribution of a zero mean and an identity matrix as a covariance matrix. After generating the instance and the targets, we presented the optimization problem of Eq. (7) to SOPOPO and to the LOQO optimization package. We repeated the above experiment for different values of k ranging from 10 through 100. For each value of k we repeated the entire experiment ten times, where in each trial we generated a new problem. We then averaged the results over the ten trials. The average CPU time consumed by the two algorithms as a function of k is depicted on the left hand side of Fig. 6. We would like to note that we have implemented SOPOPO both in Matlab and C++. We used the Matlab interface to LOQO, while LOQO itself was run in its native mode. We report results using our Matlab implementation of SOPOPO in order to eliminate possible implementation advantages. Our Matlab implementation follows the pseudo-code of Fig. 3. Nevertheless, as clearly indicated by the results, the time consumed by SOPOPO is negligible and exhibits only a very minor increase with k. In contrast, the run time of LOQO increases significantly with k. The apparent advantage of our algorithm over LOQO can be attributed to a few factors. First, LOQO is a general purpose *numerical* optimization toolkit. Its generality is clearly a two edged sword as it employs a numerical interior point method regardless of the problem on hand. Furthermore, LOQO was set to solve numerically the soft-projection problem of Eq. (7) while SOPOPO solves optimally the equivalent reduced problem of Eq. (19). To eliminate the latter mitigating factor which is in favor of SOPOPO, we repeated the same experiment as before while presenting to LOQO the reduced optimization problem rather than the original soft-projection problem. The results are depicted on the right hand side of Fig. 6. Yet again, the run time of SOPOPO is still significantly lower than LOQO for $k > 300$ and as before there is no significant increase in the run time of SOPOPO as k increases.

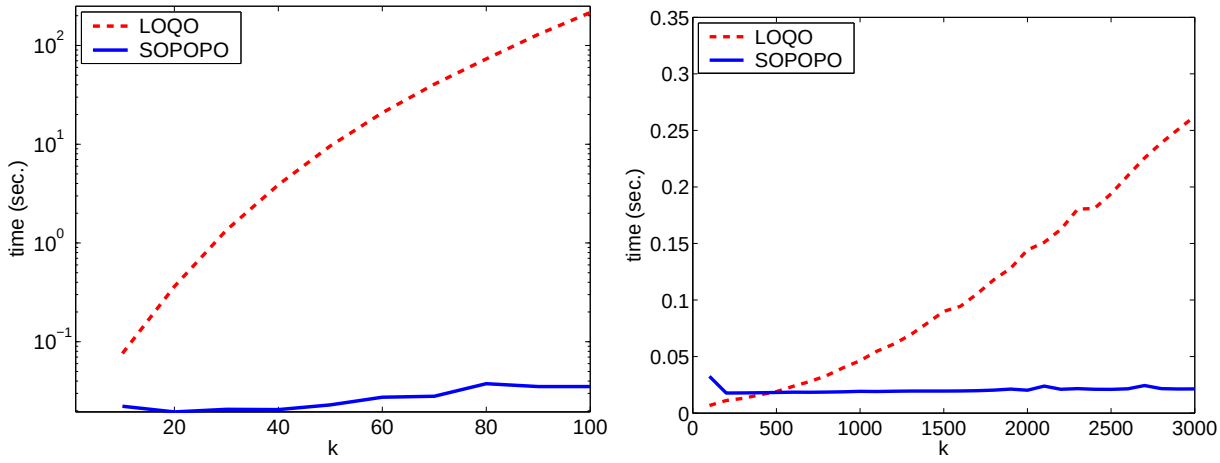

Figure 6: A comparison of the run-time of SOPOPO and LOQO on the original soft-projection problem defined in Eq. (7) (left) and on the reduced problem from Eq. (19) (right).

The second experiment compares the performance of the iterative algorithm from Fig. 5 and LOQO in the batch setting described by Eq. (6). In this experiment we generated synthetic data as follows. First, we chose the number of classes $k = |\mathcal{Y}|$ and sampled m instances from a 100-dimensional Normal distribution of a zero mean and an identity covariance matrix. We next sampled a set of vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ from the same Gaussian distribution. For each instance $\mathbf{x}^i$, we calculated the vector $\mathbf{v}^i \in \mathbb{R}^k$, whose r'th element is $\mathbf{w}_r \cdot \mathbf{x}^i$. We then set A^i to be the indices of the top $k/2$ elements of $\mathbf{v}^i$ while B^i consisted of all the rest of the elements, $[k] \setminus A^i$. For example, assume that $\mathbf{v}^i = (0.4, 4.1, 3.5, -2)$ then $A^i = \{2, 3\}$ and $B^i = \{1, 4\}$. As feedback we set $\gamma^i_a = 1$ for all $a \in A^i$ and for $b \in B^i$ we set $\gamma^i_b = 0$. In our running example, the resulting vector $\boldsymbol{\gamma}^i$ amounts to $(0, 1, 1, 0)$. Finally, we set $\mathbf{E}(\boldsymbol{\gamma}^i) = \{E^i\}$, where $E^i = A \times B$, and the value of σ was always 1. We repeated the above process for different values of k ranging from 20 through 100. The number of examples was fixed to be $10k$ and thus ranged from 200 through 1000. The value of C was set to be $1/m$. In each experiment we terminated the wrapper procedure described in Fig. 5 when the gap between the primal and dual objective functions went below 0.01. We first tried to execute LOQO with the original optimization problem described in Eq. (6). However, the resulting optimization problem was too large for LOQO to manage in a reasonable time, even for the smallest problem ($k = 20$). Our iterative algorithm solves such small problems in less than a second. To facilitate a more meaningful comparison, we used the techniques described in Sec. 3 and replaced the original optimization problem from Eq. (6) with the following reduced problem,

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & -\frac{1}{2}\sum_{r=1}^{k}\left\|\sum_{i:r\in A^i}\alpha^i_r\,\mathbf{x}^i - \sum_{i:r\in B^i}\beta^i_r\,\mathbf{x}^i\right\|^2 + \sum_{r=1}^{k}\left(\sum_{i:r\in A^i}\alpha^i_r\,\gamma^i_r - \sum_{i:r\in B^i}\beta^i_r\,\gamma^i_r\right) \\ \text{s.t.} \quad & \forall\, i\in[m]\ :\ \forall\, a\in A^i,\ \ \alpha^i_a \geq 0 \ \text{ and } \ \forall\, b\in B^i,\ \ \beta^i_b \geq 0 \\ & \forall\, i\in[m]\ :\ \sum_{a\in A^i}\alpha^i_a \ =\ \sum_{b\in B^i}\beta^i_b \ \leq\ C\ . \end{aligned} \tag{38}$$

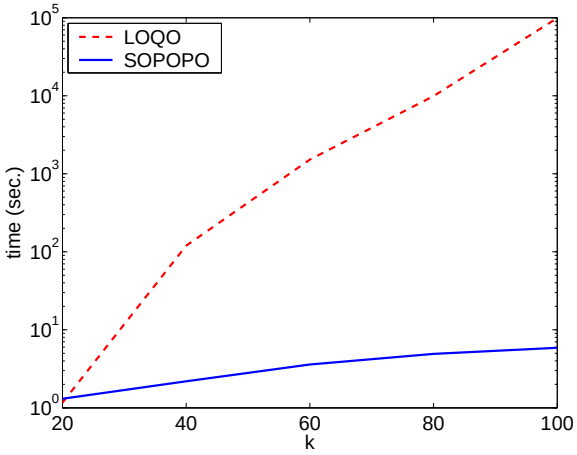

Figure 7: A comparison of the run-time in batch settings of SOPOPO and LOQO (using the reduced problem in Eq. (38)). The number of examples was set to be 10 times the number of labels (denoted k) in each problem.

By presenting the reduced problem given in Eq. (38) to LOQO, we injected quite a bit of prior knowledge that made the task manageable for LOQO. The derivation of the above reduced problem is given in appendix C. The results are summarized in Fig. 7. As clearly can be seen from the graph, our iterative algorithm outperforms LOQO, in particular as the size of the problem increases. Due to the nature of the decomposition procedure, our running time is no longer independent of the value of k as the number of graphs grows with k. Nonetheless, even for $k = 100$ the run time of SOPOPO's wrapper does not exceed 4 seconds. These promising results emphasize the viability of our approach for large scale optimization problems.

The last experiment underscores an interesting property of our iterative algorithm. In this experiment we have used the same data as in the previous experiment with $k = 100$ and $m = 1000$. After each iteration of the algorithm, we examined both the increase in the dual objective after the update and the difference between the primal and dual values. The results are shown in Fig. 8. The graphs exhibit a phenomena reminiscent of a phase transition. After about 1000 iterations, which is also the number of examples, the increase in the dual objective becomes miniscule. This phase transition is also exhibited for other choices of m, k and C. Note in addition that as the number of epochs increases, the increase of the dual objective becomes very small relatively to the duality gap. It is common to use the increase of the dual objective as a stopping criterion and the last experiment indicates that this criterion does not necessarily imply convergence. We leave further investigation of these phenomena to future research.

We would like to conclude this section with a short discussion which contrasts our approach with previous algorithms. Previous large margin approaches for label ranking associate a unique slack variable with each constraint which is induced by a pair of labels. See for example (Elisseeff and Weston, 2001) and the SVM-light implementation of label ranking (Joachims, 2002). Thus, using the terminology of this paper, these methods employ the overly simple all-pair decomposition (see Fig. 2). Using the all-pair decomposition, the label ranking problem is reduced to a binary classification problem. Indeed, the soft projection problem can be solved analytically and our wrap-

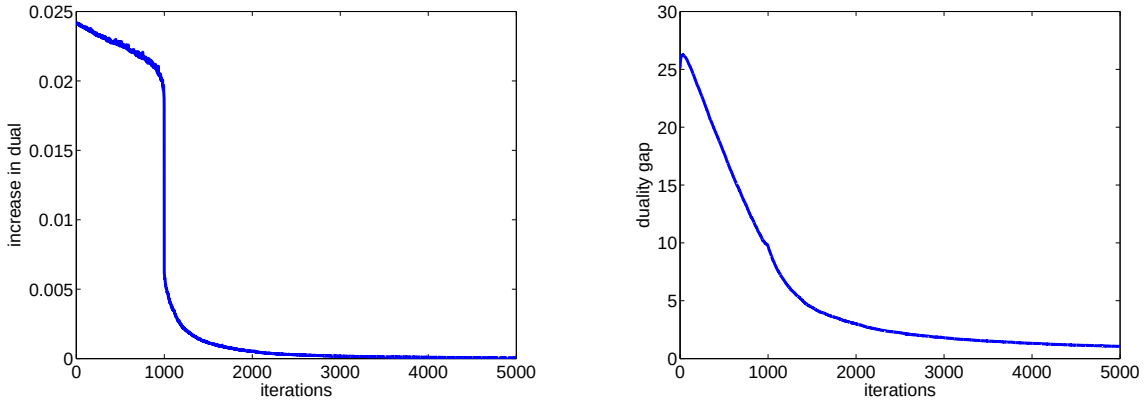

Figure 8: The increase in the dual objective (left) and the primal-dual gap (right) as a function of the number of iterations of the iterative algorithm in Fig. 5.

per algorithm from Fig. 5 is equivalent to the SOR algorithm for binary classification described in (Mangasarian and Musicant, 1999). The practical performance of the SOR algorithm for binary classification was extensively studied by Mangasarian and others. One of the main contributions of this paper is a general and flexible algorithmic framework for label ranking which can be carried with more complex decompositions. Moreover, trying to import one of the previously studied approach to our setting is difficult. A main obstacle is attributed to the fact that the set of feasible solutions for the dual problem must satisfy the constraint $\sum_a \alpha_a = \sum_b \beta_b \leq C$. Thus, a sequential minimization algorithm must update at least 4 dual variables on each iteration in order to preserve the feasibility of the dual solution. Therefore the SMO algorithm of Platt (1998) is not easily applicable to our setting. The SOPOPO algorithm suggests an efficient alternative by updating atomically all the dual variables of each sub-graph.

6. Discussion

We described an algorithmic framework for label ranking. Each iteration of our algorithm is based on SOPOPO, a fast procedure for soft projection onto a single polyhedron. There are several possible extensions of the work presented in this paper. One of them is further generalization of SOPOPO to more complex polyhedral constraints. Recall that SOPOPO is designed for projecting onto a polyhedron which is defined according to a complete bipartite graph. The generalization of SOPOPO to decompositions consisting of k-partite graphs is one particular interesting task. Another type of polyhedra that naturally emerges is regression problems with multiple outputs. In this setting, we would like the predicted differences $f_r(\mathbf{x}) - f_s(x)$ to be as close as possible to the target differences $\gamma_r - \gamma_s$, possibly up to an insensitivity term ϵ. This problem can be formalized by replacing the constraint $f_r(\mathbf{x}) - f_s(\mathbf{x}) \geq \gamma_r - \gamma_s - \xi$ with the constraint $|(f_r(\mathbf{x}) - f_s(\mathbf{x})) - (\gamma_r - \gamma_s)| \leq \epsilon + \xi$. Yet another interesting direction is the applicability of SOPOPO to online learning ranking (Crammer and Singer, 2005) where each online update is performed efficiently using SOPOPO. The phase transition phenomenon underscored in our experiments surfaces the important issue of generalization properties of our algorithm. In particular, the fact that increases in the value of dual become

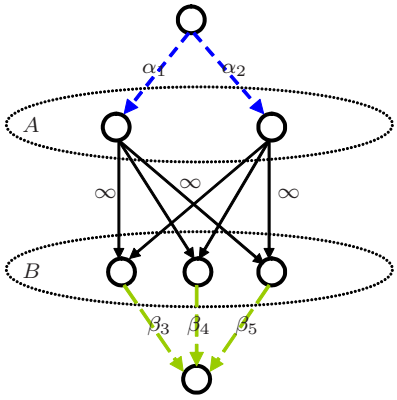

Figure 9: An illustration of the construction of a flow graph for $A = \{1, 2\}$ and $B = \{3, 4, 5\}$.

miniscule suggests the usage of early stopping so long as the prediction accuracy does not degrade. Finally, we plan to work on real world applications of SOPOPO to tasks such as category ranking for text documents.

Appendix A. The equivalence between the dual problems in Eq. (18) and Eq. (13)

In this appendix we prove that the solutions of the problem in Eq. (18) and our original dual problem from Eq. (13) are equivalent. (For an alternative derivation see also (Fung et al., 2006)). To do so, it suffices to show that for each feasible solution of the reduced problem there exists an equivalent feasible solution of the original problem and vice versa. Clearly, given $\boldsymbol{\tau}$ which satisfies the constraints imposed by Eq. (13), defining $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as given by Eq. (14) would satisfy the constraints of Eq. (18) and furthermore $D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = D(\boldsymbol{\tau})$. Denoting the optimal solution of Eq. (13) by $\boldsymbol{\tau}^\star$ and that of Eq. (18) by $(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star)$, we immediately get that $D(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star) \geq D(\boldsymbol{\tau}^\star)$. We are thus left to show that for each feasible solution $\boldsymbol{\alpha}, \boldsymbol{\beta}$ there exists a feasible solution $\boldsymbol{\tau}$ such that $D(\boldsymbol{\tau}) = \mathcal{D}(\boldsymbol{\alpha}, \boldsymbol{\beta})$. This reverse mapping is non-trivial and there does not exist a closed form description of the mapping from $\boldsymbol{\alpha}, \boldsymbol{\beta}$ to $\boldsymbol{\tau}$. The existence of such a mapping is provided in Lemma 7 below which uses the duality of max-flow and min-cut. Lemma 7 immediately implies that $D(\boldsymbol{\tau}^\star) \geq D(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star)$. In summary, we have shown that both $D(\boldsymbol{\tau}^\star) \leq D(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star)$ and $D(\boldsymbol{\tau}^\star) \geq D(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star)$ holds and therefore $D(\boldsymbol{\tau}^\star) = D(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star)$.

Lemma 7 *Let $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ be a feasible solution of the reduced problem given in Eq. (18). Then, there exists a feasible solution $\boldsymbol{\tau}$ of the original problem (Eq. (13)) such that $D(\boldsymbol{\tau}) = D(\boldsymbol{\alpha}, \boldsymbol{\beta})$.*

Proof The proof is based on the duality of max-flow and min-cut (see for example Cormen et al. (1990)). Given a feasible solution $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ defined over the sets A and B we construct a directed graph (V', E'). The set of nodes of the graph consists of the original nodes defined by the sets A and B and two additional nodes s which serves as a source and t which is a sink, $V' = A \cup B \cup \{s, t\}$. In addition to the original edges of the bipartite graph supported by A and B we add edges from s to all the nodes in A and from all the nodes in B to t and thus $E' = (A \times B) \cup (\{\mathrm{s}\} \times A) \cup (B \times \{\mathrm{t}\})$. Each edge $e \in E'$ is associated with a capacity value $c(e)$. For each $e \in A \times B$ we define $c(e) = \infty$.

For each edge of the form (s, a) where $a \in A$ we define $c(e) = \alpha_a$ and analogously for (b, t) where $b \in B$ we set $c(e) = \beta_b$. An illustration of the construction is given in Fig. 9 where $A = \{1, 2\}$ and $B = \{3, 4, 5\}$. We are now going to define a flow problem for (V', E'). We show in the sequel that *maximal* flow in the graph above defines a feasible solution for the original optimization problem. Furthermore, by using the max-flow min-cut duality, we also show that the value attained by the induced solution coincides with the value of the reduced optimization problem for $(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

A flow for the graph above is an assignment of non-negative values to edges, $\mathcal{F} : E' \rightarrow \mathbb{R}_+$, which satisfies

$$\begin{aligned} &(i) \quad \forall (r, v) \in E',\ \mathcal{F}(\,(r, v)\,) \leq \mathrm{c}(r, v) \\ &(ii) \quad \forall v \in V',\ \sum_{r:(r,v)\in E'} \mathcal{F}(\,(r, v)\,) = \sum_{r:(v,r)\in E'} \mathcal{F}(\,(v, r)\,)\ . \end{aligned} \tag{39}$$

The value of a flow function is defined as the total flow outgoing the source,

$$\mathrm{val}(\mathcal{F}) \ = \sum_{r:(\mathrm{s},r)\in E'} \mathcal{F}(\,(\mathrm{s}, r)\,)\ .$$

Let $\mathcal{F}^\star$ denote the flow attaining the maximal value among all possible flows, that is $\mathrm{val}(\mathcal{F}^\star) \geq \mathrm{val}(\mathcal{F})$. We next prove that $\mathrm{val}(\mathcal{F}^\star) = \sum_{a\in A} \alpha_a$. To do so we use the max-flow min-cut duality theorem. This theorem states that the value of the maximal flow equals the value of the minimal cut of a graph. Formally, a cut of the graph is a subset $S \subset V'$ such that $\mathrm{s} \in S$ and $\mathrm{t} \notin S$. The value of a cut is defined as the total *capacity* of edges outgoing from S to $V' \setminus S$,

$$\mathrm{val}(S) \ = \sum_{(r,v)\in S\times(V'\setminus S)\cap E'} \mathrm{c}(r, v)\ .$$

A cut is said to be minimal if its value does not exceed the value of any other cut of the graph. The value of the cut $S = \{\mathrm{s}\}$ is equal to $\sum_{y\in A} \alpha_y$. We now show that S is a minimal cut. We note in passing that while there might exist other cuts attaining the minimum value, for our purpose it suffices to show that $S = \{s\}$ is a minimal cut. Let S' be a cut different from S. Clearly, if $\mathrm{val}(S') = \infty$ then S' cannot be minimal. We thus can safely assume that $\mathrm{val}(S') < \infty$. If there exists a node $a \in A \cap S'$ then all the nodes in B must also reside in S'. Otherwise, there exists an edge (a, b) of an infinite capacity which crosses the cut and $\mathrm{val}(S') = \infty > \mathrm{val}(S)$. Since t cannot be in S' we get that for each $b \in B$, the edge (b, t) crosses the cut and therefore the value of the cut is at least $\sum_{b\in B} \beta_b = \sum_{a\in A} \alpha_a$. If on the other hand $A \cap S' = \emptyset$ then all the edges from s to the nodes in A cross the cut. Therefore, $\mathrm{val}(S)$ is again at least $\sum_{y\in A} \alpha_y$. We have thus shown that $S = \{\mathrm{s}\}$ is a minimal cut of the flow graph.

From the duality theorem of max-flow and min-cut we get that there exists a minimal flow $\mathcal{F}^\star$ such that $\mathrm{val}(\mathcal{F}^\star) = \sum_{a\in A} \alpha_a$. Since each outgoing edge from s hits a different node in A, we must have that $\mathcal{F}^\star(\,(\mathrm{s}, a)\,) = \alpha_a$ in order to reach the optimal flow value. Similarly, for each $b \in B$ we get that $\mathcal{F}^\star(\,(b, \mathrm{t})\,) = \beta_b$. We now set $\tau_{a,b} = \mathcal{F}^\star(\,(a, b)\,)$ for each $(a, b) \in A \times B$. Since a proper flow associates a non-negative value with each edge we obtain that $\tau_{a,b} \geq 0$. From the conservation of flow we get that,

$$\alpha_a = \mathcal{F}^\star(\,(\mathrm{s}, a)\,) = \sum_{b\in B} \mathcal{F}^\star(\,(a, b)\,) = \sum_{b\in B} \tau_{a,b}\ ,$$

and

$$\beta_b = \mathcal{F}^\star(\,(b,\mathrm{t})\,) = \sum_{a\in A} \mathcal{F}^\star(\,(a,b)\,) = \sum_{a\in A} \tau_{a,b} \ .$$

Thus, this construction of $\boldsymbol{\tau}$ from the optimal flow satisfies the equalities given in Eq. (14). By construction, each node $a \in A$ has one incoming edge (s,a) and outgoing edges to all nodes in B. Thus, the flow conservation requirement of Eq. (39) again implies that

$$C \ \geq \ \sum_{a\in A} \mathcal{F}^\star(\,(\mathrm{s},a)\,) = \sum_{a\in A, b\in B} \mathcal{F}^\star(\,(a,b)\,) = \sum_{a\in A, b\in B} \tau_{a,b} \ .$$

Therefore, $\boldsymbol{\tau}$ adheres with the constraints of Eq. (13). In summary, we have constructed a feasible solution for the original constrained optimization problem which is consistent with the definitions of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Therefore, $D(\boldsymbol{\tau}) = D(\boldsymbol{\alpha},\boldsymbol{\beta})$ as required. ■

Appendix B. Technical Proofs

Proof of Lemma 3

Throughout the proof we assume that the elements of the vector $\boldsymbol{\mu}$ are sorted in a non-ascending order, namely, $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_p$. Recall that the definition of $\rho(z,\boldsymbol{\mu})$ is,

$$\rho(z,\boldsymbol{\mu}) \ = \ \max\left\{ j \in [p] \ : \ \mu_j - \frac{1}{j}\left(\sum_{r=1}^{j} \mu_r - z\right) > 0 \right\} \ .$$

For brevity, we refer to $\rho(z,\boldsymbol{\mu})$ simply as ρ. Denote by $\boldsymbol{\alpha}^\star$ the optimal solution of the constrained optimization problem of Eq. (20) and let

$$\rho^\star = \max\{j \ : \ \alpha^\star_j > 0\} \ .$$

From Eq. (23) we know that $\alpha^\star_r = \mu_r - \theta^\star > 0$ for $r \leq \rho^\star$ where

$$\theta^\star = \frac{1}{\rho^\star}\left(\sum_{j=1}^{\rho^\star} \mu_j - z\right) \ ,$$

and therefore $\rho \geq \rho^\star$. We thus need to prove that $\rho = \rho^\star$. Assume by contradiction that $\rho > \rho^\star$. Let us denote by $\boldsymbol{\alpha}$ the vector induced by the choice of ρ, that is, $\alpha_r = 0$ for $r > \rho$ and $\alpha_r = \mu_r - \theta$ for $r \leq \rho$, where,

$$\theta \ = \ \frac{1}{\rho}\left(\sum_{j=1}^{\rho} \mu_j - z\right) \ .$$

From the definition of ρ, we must have that $\alpha_\rho = \mu_\rho - \theta > 0$. Therefore, since the elements of $\boldsymbol{\mu}$ are sorted in a non-ascending order, we get that $\alpha_r = \mu_r - \theta > 0$ for all $r \leq \rho$. In addition, the choice of θ implies that $\|\boldsymbol{\alpha}\|_1 = z$. We thus get that $\boldsymbol{\alpha}$ is a feasible solution as it satisfies the

constraints of Eq. (20). Examining the objective function attained at $\boldsymbol{\alpha}$ we get that,

$$\begin{aligned}
\|\boldsymbol{\alpha}-\boldsymbol{\mu}\|^2 &= \sum_{r=1}^{\rho^\star}\theta^2 + \sum_{r=\rho^\star+1}^{\rho}\theta^2 + \sum_{r=\rho+1}^{p}\mu_r^2 \\
&< \sum_{r=1}^{\rho^\star}\theta^2 + \sum_{r=\rho^\star+1}^{\rho}\mu_r^2 + \sum_{r=\rho+1}^{p}\mu_r^2 \\
&= \sum_{r=1}^{\rho^\star}\theta^2 + \sum_{r=\rho^\star+1}^{p}\mu_r^2 \ ,
\end{aligned}$$

where to derive the inequality above we used the fact that $\mu_r - \theta > 0$ for all $r \leq \rho$. We now need to analyze two cases depending on whether $\theta^\star$ is greater than θ or not. If $\theta^\star \geq \theta$ than we can further bound $\|\boldsymbol{\alpha}-\boldsymbol{\mu}\|^2$ from above as follows,

$$\|\boldsymbol{\alpha}-\boldsymbol{\mu}\|^2 \ < \ \sum_{r=1}^{\rho^\star}\theta^2 + \sum_{r=\rho^\star+1}^{p}\mu_r^2 \ \leq \ \sum_{r=1}^{\rho^\star}(\theta^\star)^2 + \sum_{r=\rho^\star+1}^{p}\mu_r^2 \ = \ \|\boldsymbol{\alpha}^\star-\boldsymbol{\mu}\|^2 \ ,$$

which contradicts the optimality of $\boldsymbol{\alpha}^\star$. We are thus left to show that the case $\theta > \theta^\star$ also leads to a contradiction. We do so by constructing a vector $\tilde{\boldsymbol{\alpha}}$ from $\boldsymbol{\alpha}^\star$. We show that this vector satisfies the constraints of Eq. (20) hence it is a feasible solution. Finally, we show that the objective function attained by $\tilde{\boldsymbol{\alpha}}$ is strictly smaller than that of $\boldsymbol{\alpha}^\star$. We define the vector $\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^k$ as follows,

$$\tilde{\alpha}_r = \begin{cases} \alpha^\star_{\rho^\star} - \epsilon & r = \rho^\star \\ \epsilon & r = \rho^\star + 1 \\ \alpha^\star_r & \text{otherwise} \end{cases} \ ,$$

where $\epsilon = \frac{1}{2}(\mu_{\rho^\star+1} - \theta^\star)$. Since we assume that $\theta > \theta^\star$ and $\rho > \rho^\star$ we know that $\alpha_{\rho^\star+1} = \mu_{\rho^\star+1} - \theta > 0$ which implies that

$$\tilde{\alpha}_{\rho^\star+1} = \frac{1}{2}(\mu_{\rho^\star+1} - \theta^\star) > \frac{1}{2}(\mu_{\rho^\star+1} - \theta) = \frac{1}{2}\alpha_{\rho^\star+1} > 0 \ .$$

Furthermore, we also get that,

$$\tilde{\alpha}_{\rho^\star} \ = \ \mu_{\rho^\star} - \frac{1}{2}\mu_{\rho^\star+1} - \frac{1}{2}\theta^\star \ > \ \frac{1}{2}\left(\mu_{\rho^\star+1} - \theta\right) \ = \ \frac{1}{2}\alpha_{\rho^\star+1} \ > \ 0 \ .$$

In addition, by construction we get that the rest of components of $\tilde{\boldsymbol{\alpha}}$ are also non-negative. Our construction also preserves the norm, that is $\|\tilde{\boldsymbol{\alpha}}\|_1 = \|\boldsymbol{\alpha}^\star\|_1 = z$. Thus, the vector $\tilde{\boldsymbol{\alpha}}$ is also a feasible solution for the set of constraints defined by Eq. (20). Alas, examining the difference in the objective functions attained by $\tilde{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}^\star$ we get,

$$\begin{aligned}
\|\boldsymbol{\alpha}^\star-\boldsymbol{\mu}\|^2 - \|\tilde{\boldsymbol{\alpha}}-\boldsymbol{\mu}\|^2 &= (\theta^\star)^2 + \mu^2_{\rho^\star+1} - \left((\theta^\star+\epsilon)^2 + (\mu_{\rho^\star+1}-\epsilon)^2\right) \\
&= 2\epsilon\underbrace{(\mu_{\rho^\star+1} - \theta^\star)}_{=2\epsilon} - 2\epsilon^2 \ = \ 2\,\epsilon^2 \ > \ 0 \ .
\end{aligned}$$

We thus obtained the long desired contradiction which concludes the proof. ■

Proof of Thm. 4
Plugging the value of the optimal solution $\boldsymbol{\alpha}$ from Eq. (23) into the objective $\|\boldsymbol{\alpha} - \boldsymbol{\mu}\|^2$ and using Lemma 3 give that,

$$g(z;\boldsymbol{\mu}) \;=\; \frac{1}{\rho(z;\boldsymbol{\mu})}\left(\sum_{r=1}^{\rho(z;\boldsymbol{\mu})}\mu_r - z\right)^2 + \sum_{r=\rho(z;\boldsymbol{\mu})+1}\mu_r^2 \;\;,$$

where, to remind the reader, the number of strictly positive $\boldsymbol{\alpha}$'s is,

$$\rho(z;\boldsymbol{\mu}) = \max\left\{\rho : \mu_\rho - \frac{1}{\rho}\left(\sum_{r=1}^{\rho}\mu_r - z\right) \geq 0\right\} \;\;.$$

Throughout the proof $\boldsymbol{\mu}$ is fixed and known. We therefore abuse our notation and use the shorthand $\rho(z)$ for $\rho(z;\boldsymbol{\mu})$. Recall that $\boldsymbol{\mu}$ is given in a non-ascending order, $\mu_{i+1} \leq \mu_i$ for $i \in [p-1]$. Therefore, we get that

$$\begin{aligned} z_{i+1} \;&=\; \sum_{r=1}^{i+1}\mu_r - (i+1)\mu_{i+1} \;=\; \sum_{r=1}^{i}\mu_r + \mu_{i+1} - \mu_{i+1} - i\,\mu_{i+1} \\ &=\; \sum_{r=1}^{i}\mu_r - i\,\mu_{i+1} \;\geq\; \sum_{r=1}^{i}\mu_r - i\,\mu_i \;=\; z_i \;\;. \end{aligned}$$

Thus, the sequence $z_1, z_2, \ldots, z_p$ is monotonically non-decreasing and the intervals $[z_i, z_{i+1})$ are well defined. The definition of $\rho(z)$ implies that for all $z \in [z_i, z_{i+1})$ we have $\rho(z) = \rho(z_i) = i$. Hence, the value of $g(z;\boldsymbol{\mu})$ for each $z \in [z_i, z_{i+1})$ is,

$$g(z;\boldsymbol{\mu}) \;=\; \frac{1}{i}\left(\sum_{r=1}^{i}\mu_r - z\right)^2 + \sum_{r=i+1}^{p}\mu_r^2 \;\;.$$

We have thus established the fact that $g(z;\boldsymbol{\mu})$ is a quadratic function in each interval (z_i, z_{i+1}) and in particular it is continuous in each such sub-interval. To show that g is continuous in $[0, C]$ we need to examine all of its knots z_i. Computing the left limit and the right limit of g at each knot we get that,

$$\begin{aligned} \lim_{z\downarrow z_i} g(z;\boldsymbol{\mu}) \;&=\; \lim_{z\downarrow z_i}\frac{1}{i}\left(\sum_{r=1}^{i}\mu_r - z\right)^2 + \sum_{r=i+1}^{p}\mu_r^2 \\ &=\; \frac{1}{i}\left(\sum_{r=1}^{i}\mu_r - \sum_{r=1}^{i}\mu_r + i\mu_i\right)^2 + \sum_{r=i+1}^{p}\mu_r^2 \\ &=\; i\mu_i^2 + \sum_{r=i+1}^{p}\mu_r^2 \;\;, \end{aligned}$$

and

$$\begin{aligned}\lim_{z\uparrow z_i} g(z;\boldsymbol{\mu}) &= \lim_{z\uparrow z_i} \frac{1}{i-1}\left(\sum_{r=1}^{i-1}\mu_r - z\right)^2 + \sum_{r=i}^{p}\mu_r^2 \\ &= \frac{1}{i-1}\left(\sum_{r=1}^{i-1}\mu_r - \sum_{r=1}^{i}\mu_r + i\mu_i\right)^2 + \sum_{r=i}^{p}\mu_r^2 \\ &= (i-1)\mu_i^2 + \sum_{r=i}^{p}\mu_r^2 \;=\; i\mu_i^2 + \sum_{r=i+1}^{p}\mu_r^2 \;\;.\end{aligned}$$

Therefore, $\lim_{z\downarrow z_i} g(z;\boldsymbol{\mu}) = \lim_{z\uparrow z_i} g(z;\boldsymbol{\mu})$ and g is indeed continuous. The continuity of the derivative of g is shown by using the same technique of examining the right and left limits at each knot z_i for the function,

$$g'(z;\boldsymbol{\mu}) = \frac{2}{i}\left(z - \sum_{r=1}^{i}\mu_r\right) \;\;.$$

Finally, we use the fact that a continuously differentiable function is convex iff its derivative is monotonically non-decreasing. Since g is quadratic in each segment $[z_i, z_{i+1}]$, g' is indeed monotonically non-decreasing in each segment. Furthermore, from the continuity of g' we get that g' is monotonically non-decreasing on the entire interval $[0, C]$. Thus, g is convex on $[0, C]$. ■

Appendix C. Derivation of Eq. (38)

In this section we derive conversion of the optimization problem from Eq. (6) to its reduced form given in Eq. (38). In Sec. 4 (Eq. (30)) we derived the dual of Eq. (6). Assuming that for each example, $\mathbf{E}(\boldsymbol{\gamma}^i) = \{A^i \times B^i\}$, and using the definitions of $\mathbf{a}^{i,j}$, $b^{i,j}$, and $\bar{\mathbf{w}}$ from Sec. 4, we can rewrite the dual of Eq. (6) as

$$\begin{aligned}\max_{\boldsymbol{\tau}} \quad & -\frac{1}{2}\sum_{r=1}^{k}\|\mathbf{w}_r\|^2 \;+\; \sum_{i=1}^{m}\sum_{a\in A^i}\sum_{b\in B^i}\lambda_{a,b}^i\left(\gamma_a^i - \gamma_b^i\right) \\ \text{s.t.} \;& \forall\, i\in[m] \;:\; \forall\,(a,b)\in A^i\times B^i,\;\; \lambda_{a,b}^i \geq 0 \\ & \forall\, i\in[m] \;:\; \sum_{(a,b)\in A^i\times B^i}\lambda_{a,b}^i \leq C \quad ,\end{aligned} \tag{40}$$

where

$$\mathbf{w}_r \;=\; \sum_{i:r\in A^i}\sum_{b\in B^i}\lambda_{r,b}^i\mathbf{x}^i - \sum_{i:r\in B^i}\sum_{a\in A^i}\lambda_{a,r}^i\mathbf{x}^i \;\;. \tag{41}$$

For each $a \in A^i$ define,

$$\alpha_a^i \;=\; \sum_{b\in B^i}\lambda_{a,b}^i \;\;, \tag{42}$$

and similarly, for each $b \in B^i$ define,

$$\beta_b^i \;=\; \sum_{a\in A^i}\lambda_{a,b}^i \;\;. \tag{43}$$

Using these definitions, we can rewrite Eq. (41) as,

$$\mathbf{w}_r \;=\; \sum_{i:r\in A^i} \alpha_r^i \mathbf{x}^i - \sum_{i:r\in B^i} \beta_r^i \mathbf{x}^i \;\;.$$

Therefore, the dual objective can be rewritten as,

$$D \;=\; -\frac{1}{2}\sum_{r=1}^{k}\left\| \sum_{i:r\in A^i} \alpha_r^i\,\mathbf{x}^i \;-\; \sum_{i:r\in B^i} \beta_r^i\,\mathbf{x}^i \right\|^2 \;+\; \sum_{r=1}^{k}\left(\sum_{i:r\in A^i} \alpha_r^i\,\gamma_r^i - \sum_{i:r\in B^i} \beta_r^i\,\gamma_r^i \right) \;\;.$$

As in Sec. 3, we need to enforce the additional constraints on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$,

$$\begin{aligned} &\forall\, i \in [m] \;:\forall\, a \in A^i,\;\; \alpha_a^i \geq 0 \;\text{ and }\; \forall\, b \in B^i,\;\; \beta_b^i \geq 0 \\ &\forall\, i \in [m] \;:\; \sum_{a\in A^i} \alpha_a^i \;=\; \sum_{b\in B^i} \beta_b^i \;\leq\; C \;\;. \end{aligned}$$

Combining the dual definition with the above constraints gives the reduced problem from Eq. (38).

Acknowledgments

We would like to thank Koby Crammer for numerous fruitful discussions. Aaron D'Souza and Vineet Gupta have played a key role in motivating this line of research. Thanks also to Yair Censor, Mayur Datar, Ofer Dekel, Phil Long, Michael fink, and Peter Norvig for their comments and feedback. The work of Shai Shalev-Shwartz was supported by the Israeli Science Foundation under grant no. 522/04.

References

S. Agarwal and P. Niyogi. Stability and generalization of bipartite ranking algorithms. In *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory*, pages 32–47, 2005.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Y. Censor and S.A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, New York, NY, USA, 1997.

S. Clemenon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. In *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory*, 2005.

W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Jornal of Machine Learning Research*, 2:265–292, 2001.

K. Crammer and Y. Singer. A new family of online algorithms for category ranking. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

K. Crammer and Y. Singer. Online ranking by projecting. *Neural Computation*, 17(1), 2005.

O. Dekel, C. Manning, and Y. Singer. Log-linear models for label ranking. In *Advances in Neural Information Processing Systems 16*, 2003.

A. Elisseeff and J. Weston. A kernel method for multi-labeled classification. In *Advances in Neural Information Processing Systems 14*, 2001.

G. Fung, R. Rosales, and B. Krishnapuram. Learning rankings via convex hull separation. In *Advances in Neural Information Processing Systems 18*, 2006.

R. Herbrich, T. Graepel, and K. Obermayer. Large marging rank boundaries for ordinal regression. In A. Smola, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 2000.

C. Hildreth. A quadratic programming procedure. *Naval Research Logistics Quarterly*, 4:79–85, 1957. Erratum, ibidem, p.361.

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

C.-J. Lin. A formal analysis of stopping criteria of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 13(5):1045–1052, Sept. 2002.

O. Mangasarian and D. Musicant. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10, 1999.

J. C. Platt. Fast training of Support Vector Machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

C. Rudin, C. Cortes, M. Mohri, and R.E. Schapire. Margin-based ranking and boosting meet in the middle. In *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory*, pages 63–78, 2005.

R.J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Mthods and Software*, 12:451–484, 1999.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of the Seventh European Symposium on Artificial Neural Networks*, April 1999.