Journal of the Operations Research Society of Japan
Vol. 58, No. 4, October 2015, pp. 394–409

ROW AND COLUMN GENERATION ALGORITHMS FOR MINIMUM MARGIN MAXIMIZATION OF RANKING PROBLEMS

Yoichi Izunaga
University of Tsukuba

Keisuke Sato
Railway Technical Research Institute

Keiji Tatsumi
Osaka University

Yoshitsugu Yamamoto
University of Tsukuba

(Received October 24, 2014; Revised August 20, 2015)

Abstract We consider the ranking problem of learning a ranking function from the data set of objects each of which is endowed with an attribute vector and a ranking label chosen from the ordered set of labels. We propose two different formulations: primal problem, primal problem with dual representation of normal vector, and then propose to apply the kernel technique to the latter formulation. We also propose algorithms based on the row and column generation in order to mitigate the computational burden due to the large number of objects.

Keywords: Optimization, ranking problem, support vector machine, multi-class classification, kernel technique, dual representation

1. Introduction

This paper is concerned with a multi-class classification problem of n objects, each of which is endowed with an m-dimensional *attribute vector* $\boldsymbol{x}^i = (x^i_1, x^i_2, \dots, x^i_m)^\top \in \mathbb{R}^m$ and a *label* ℓ_i. The underlying statistical model assumes that object i receives label k, i.e., $\ell_i = k$, when the latent variable y_i determined by

$$y_i = \boldsymbol{w}^\top \boldsymbol{x}^i + \varepsilon^i = \sum_{j=1}^m w_j x^i_j + \varepsilon^i$$

falls between two thresholds p_k and p_{k+1}, where ε^i represents a random noise whose probabilistic property is not known. Namely, attribute vectors of objects are loosely separated by hyperplanes $H(\boldsymbol{w}, p_k) = \{\, \boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{w}^\top \boldsymbol{x} = p_k \,\}$ for $k = 1, 2, \dots, l$ which share a common normal vector $\boldsymbol{w}$, then each object is given a label according to the layer it is located in. Note that neither y_i's, w_j's nor p_k's are observable. Our problem is to find the normal vector $\boldsymbol{w} \in \mathbb{R}^m$ as well as the thresholds $p_1, p_2, \dots, p_l$ that best fit the input data $\{\, (\boldsymbol{x}^i, \ell_i) \mid i = 1, 2, \dots, n \,\}$.

This problem is known as the *ranking problem* and frequently arises in social sciences and operations research. See, for instance Crammer and Singer [2], Herbrich *et al.* [3], Liu [4], Shashua and Levin [6] and Chapter 8 of Shawe-Taylor and Cristianini [7]. It is a variation of the multi-class classification problem, for which several learning algorithms of the *support vector machine* (*SVM* for short) have been proposed. We refer the reader to Chapters 4.1.2 and 7.1.3 of Bishop [1], Chapter 10.10 of Vapnik [9] and Tatsumi *et al.* [8] and references therein. What distinguishes the problem from other multi-class classification problems is that the identical normal vector should be shared by all the separating hyperplanes. In this paper based on the formulation *fixed margin strategy* by Shashua and Levin [6], we propose a row and column generation algorithm to maximize the minimum margin for the ranking problems.

This paper is organized as follows. We give some definitions and notation in Section 2. In Section 3, we formulate the maximization of minimum margin with the hard margin constraints and apply the dual representation of the normal vector to the formulation. In Section 4, we propose a row and column generation algorithm and prove the validity of the algorithm. In Section 5, after reviewing the kernel technique, we apply the kernel technique to the hard margin problem with the dual representation. In Section 6, 7 and 8, we broaden the discussions so far to the soft margin problem. After giving a small illustrative example in Section 9, we report the computational experiments of our algorithm in Section 10. In appendix, we discuss the monotonicity of the separating curves.

2. Definitions and Notation

Throughout the paper $N = \{1, 2, \dots, i, \dots, n\}$ denotes the set of n objects and $\boldsymbol{x}^i = (x_1^i, x_2^i, \dots, x_m^i)^\top \in \mathbb{R}^m$ denotes the attribute vector of object i. The predetermined set of labels is $L = \{0, 1, \dots, k, \dots, l\}$ and the label assigned to object i is denoted by ℓ_i. Let $N(k) = \{\, i \in N \mid \ell_i = k \,\}$ be the set of objects with label $k \in L$, and for notational convenience we write $n(k) = |N(k)|$ for $k \in L$, and $N(k..k') = N(k) \cup N(k+1) \cup \dots \cup N(k')$ for $k, k' \in L$ such that $k < k'$. For succinct notation we define

$$X = \left[\begin{array}{ccc} \cdots & \boldsymbol{x}^i & \cdots \end{array} \right]_{i \in N} \in \mathbb{R}^{m \times n} \tag{2.1}$$

$$X_W = \left[\begin{array}{c} \cdots \boldsymbol{x}^i \cdots \end{array} \right]_{i \in W} \in \mathbb{R}^{m \times |W|} \tag{2.2}$$

for $W \subseteq N$, and the corresponding Gram matrices

$$K = X^\top X \in \mathbb{R}^{n \times n}, \tag{2.3}$$

$$K_W = X_W^\top X_W \in \mathbb{R}^{|W| \times |W|}. \tag{2.4}$$

We denote the k-dimensional zero vector and the k-dimensional vector of 1's by $\mathbf{0}_k$ and $\mathbf{1}_k$, respectively. Given a subset $W \subseteq N$ and a vector $\boldsymbol{\alpha} = (\alpha_i)_{i \in W}$ we use the notation $(\boldsymbol{\alpha}_W, \mathbf{0}_{N \setminus W})$ to denote the n-dimensional vector $\bar{\boldsymbol{\alpha}}$ such that

$$\bar{\alpha}_i = \begin{cases} \alpha_i & \text{when } i \in W \\ 0 & \text{otherwise.} \end{cases}$$

3. Hard Margin Problems for Separable Case

3.1. Primal hard margin problem

Henceforth we assume that $N(k) \neq \emptyset$ for all $k \in L$ for the sake of simplicity, and adopt the notational convention that $p_0 = -\infty$ and $p_{l+1} = +\infty$. We say that an instance $\{\, (\boldsymbol{x}^i, \ell_i) \mid i \in N \,\}$ is *separable* if there exist $\boldsymbol{w} \in \mathbb{R}^m$ and $\boldsymbol{p} = (p_1, p_2, \dots, p_l)^\top \in \mathbb{R}^l$ such that

$$p_{\ell_i} < \boldsymbol{w}^\top \boldsymbol{x}^i < p_{\ell_i + 1} \quad \text{for } i \in N.$$

Clearly an instance is separable if and only if there are $\boldsymbol{w}$ and $\boldsymbol{p}$ such that

$$p_{\ell_i} + 1 \le \boldsymbol{w}^\top \boldsymbol{x}^i \le p_{\ell_i+1} - 1 \quad \text{for } i \in N.$$

For each $k \in L \setminus \{0\}$ we see that

$$\max_{i \in N(k-1)} \boldsymbol{w}^\top \boldsymbol{x}^i \le p_k - 1 < p_k < p_k + 1 \le \min_{j \in N(k)} \boldsymbol{w}^\top \boldsymbol{x}^j,$$

implying

$$\min_{j \in N(k)} \frac{\boldsymbol{w}^\top}{\|\boldsymbol{w}\|} \boldsymbol{x}^j - \max_{i \in N(k-1)} \frac{\boldsymbol{w}^\top}{\|\boldsymbol{w}\|} \boldsymbol{x}^i \ge \frac{2}{\|\boldsymbol{w}\|}.$$

Then the margin between $\{ \boldsymbol{x}^i \mid i \in N(k-1) \}$ and $\{ \boldsymbol{x}^j \mid j \in N(k) \}$ is at least $2/\|\boldsymbol{w}\|$ for $k = 2, \ldots, l$. Hence the maximization of the minimum margin is formulated as the quadratic programming

$$(H) \quad \left| \begin{array}{ll} \text{minimize} & \|\boldsymbol{w}\|^2 \\ \text{subject to} & p_{\ell_i} + 1 \le (\boldsymbol{x}^i)^\top \boldsymbol{w} \le p_{\ell_i+1} - 1 \qquad \text{for } i \in N, \end{array} \right.$$

or more explicitly with the notation introduced in Section 2

$$(H) \quad \left| \begin{array}{lll} \text{minimize} & \|\boldsymbol{w}\|^2 & \\ \text{subject to} & 1 - (\boldsymbol{x}^i)^\top \boldsymbol{w} + p_{\ell_i} \le 0 & \text{for } i \in N(1..l) \\ & 1 + (\boldsymbol{x}^i)^\top \boldsymbol{w} - p_{\ell_i+1} \le 0 & \text{for } i \in N(0..l-1). \end{array} \right.$$

The constraints therein are called the *hard margin* constraints, and we name this problem (H).

3.2. Dual representation

A close look at the primal problem (H) shows that the following property holds for an optimum solution $\boldsymbol{w}^*$. See, for example Chapter 6 of Bishop [1], Shashua and Levin [6] and Theorem 1 in Schölkopf *et al.* [5].

Lemma 3.1. *Let* $(\boldsymbol{w}^*, \boldsymbol{p}^*) \in \mathbb{R}^{m+l}$ *be an optimum solution of* (H). *Then* $\boldsymbol{w}^* \in \mathbb{R}^m$ *lies in the range space of* X, *i.e.,* $\boldsymbol{w}^* = X\boldsymbol{\lambda}$ *for some* $\boldsymbol{\lambda} \in \mathbb{R}^n$.

Proof. Let $\boldsymbol{w}_1$ be the orthogonal projection of $\boldsymbol{w}^*$ onto the range space of X and let $\boldsymbol{w}_2 = \boldsymbol{w}^* - \boldsymbol{w}_1$. Then we obtain

$$(\boldsymbol{x}^i)^\top \boldsymbol{w}^* = (\boldsymbol{x}^i)^\top (\boldsymbol{w}_1 + \boldsymbol{w}_2) = (\boldsymbol{x}^i)^\top \boldsymbol{w}_1 \quad \text{for } i \in N,$$

meaning that $(\boldsymbol{w}_1, \boldsymbol{p}^*)$ is feasible to (H), and

$$\|\boldsymbol{w}^*\|^2 = \|\boldsymbol{w}_1\|^2 + \|\boldsymbol{w}_2\|^2 \ge \|\boldsymbol{w}_1\|^2.$$

Hence by the optimality of $\boldsymbol{w}^*$ we conclude that $\boldsymbol{w}_2 = \boldsymbol{0}$. □

The representation $\boldsymbol{w} = X\boldsymbol{\lambda}$ is called the *dual representation* of the normal vector. Substituting $X\boldsymbol{\lambda}$ for $\boldsymbol{w}$ yields another formulation of the primal hard margin problem $(\bar{H})$:

$$(\bar{H}) \quad \left| \begin{array}{ll} \text{minimize} & \boldsymbol{\lambda}^\top K \boldsymbol{\lambda} \\ \text{subject to} & p_{\ell_i} + 1 \le (\boldsymbol{k}^i)^\top \boldsymbol{\lambda} \le p_{\ell_i+1} - 1 \qquad \text{for } i \in N, \end{array} \right.$$

where $(\boldsymbol{k}^i)^\top = ((\boldsymbol{x}^i)^\top \boldsymbol{x}^1, (\boldsymbol{x}^i)^\top \boldsymbol{x}^2, \ldots, (\boldsymbol{x}^i)^\top \boldsymbol{x}^n)$ is the ith row of the matrix K. Since n is typically by far larger than m, problem $(\bar{H})$ might be less interesting than problem (H). However the fact that this formulation only requires the matrix K will enable an application of the kernel technique to the problem.

4. Algorithms for Hard Margin Problems

We start with proposing an algorithm for problem (H) arising from separable instances. Note that the separability makes (H) feasible. The problem has $m + l$ of variables and by far larger n of constraints. It is very likely that a small fraction of constraints is binding at an optimum solution of the problem, i.e., a small number of support vectors is expected. Introducing a subset W, called the *working set*, of N and omitting the constraints for i not in W, we consider the relaxed problem:

$$(H(W)) \quad \left| \begin{array}{ll} \text{minimize} & \|\boldsymbol{w}\|^2 \\ \text{subject to} & p_{\ell_i} + 1 \le (\boldsymbol{x}^i)^\top \boldsymbol{w} \le p_{\ell_i+1} - 1 \qquad \text{for } i \in W. \end{array} \right.$$

If an optimum solution of problem $(H(W))$ satisfies all the constraints for $i \in N \setminus W$, it is obviously an optimum solution of (H). Therefore the following row generation algorithm will solve problem (H) when it terminates.

Algorithm RH (Row Generation Algorithm for (H))

```
Step 1 : Let $W^0$ be an initial working set and let $\nu = 0$.
Step 2 : Solve $(H(W^\nu))$ to obtain $\boldsymbol{w}^\nu$ and $\boldsymbol{p}^\nu$.
Step 3 : Let $\Delta = \{\, i \in N \setminus W^\nu \mid (\boldsymbol{w}^\nu, \boldsymbol{p}^\nu)$ violates $p_{\ell_i} + 1 \le (\boldsymbol{x}^i)^\top \boldsymbol{w} \le p_{\ell_i+1} - 1 \,\}$.
Step 4 : If $\Delta = \emptyset$, terminate.
Step 5 : Otherwise choose $\Delta^\nu \subseteq \Delta$, let $W^{\nu+1} = W^\nu \cup \Delta^\nu$, increment $\nu$ by 1 and go to Step 2.
```

Next we consider the primal hard margin problem $(\bar{H})$ with the dual representation of the normal vector. Since the dimension m of the attribute vector is much smaller than the number n of objects, it is very likely that a small number of λ_i's are positive in the dual representation $\boldsymbol{w} = X\boldsymbol{\lambda}$. Then we propose to start the algorithm with a small number of attribute vectors as W and then increment it as the computation goes on. The sub-problem to solve is

$$(\bar{H}(W)) \quad \left| \begin{array}{ll} \text{minimize} & \boldsymbol{\lambda}_W^\top K_W \boldsymbol{\lambda}_W \\ \text{subject to} & p_{\ell_i} + 1 \le (\boldsymbol{k}_W^i)^\top \boldsymbol{\lambda}_W \le p_{\ell_i+1} - 1 \quad \text{for } i \in W, \end{array} \right.$$

where $(\boldsymbol{k}_W^i)^\top$ is the row vector consisting $(\boldsymbol{x}^i)^\top \boldsymbol{x}^j$ for $j \in W$. Note that the dimension of $\boldsymbol{\lambda}_W$ varies when the size of W changes as the computation goes on.

Algorithm RC$\bar{\text{H}}$ (Row and Column Generation Algorithm for $(\bar{H})$)

```
Step 1 : Let $W^0$ be an initial working set, and let $\nu = 0$.
Step 2 : Solve $(\bar{H}(W^\nu))$ to obtain $\boldsymbol{\lambda}_{W^\nu}$ and $\boldsymbol{p}^\nu$.
Step 3 : Let $\Delta = \{\, i \in N \setminus W^\nu \mid (\boldsymbol{\lambda}_{W^\nu}, \boldsymbol{p}^\nu)$ violates $p_{\ell_i} + 1 \le (\boldsymbol{k}_{W^\nu}^i)^\top \boldsymbol{\lambda}_W \le p_{\ell_i+1} - 1 \,\}$.
Step 4 : If $\Delta = \emptyset$, terminate.
Step 5 : Otherwise choose $\Delta^\nu \subseteq \Delta$, let $W^{\nu+1} = W^\nu \cup \Delta^\nu$, increment $\nu$ by 1 and go to Step 2.
```

The following lemma shows that Algorithm RC$\bar{\text{H}}$ solves problem $(\bar{H})$ upon termination.

Lemma 4.1. *Let* $(\hat{\boldsymbol{\lambda}}_W, \hat{\boldsymbol{p}}) \in \mathbb{R}^{|W|+l}$ *be an optimum solution of* $(\bar{H}(W))$. *If*

$$\hat{p}_{\ell_i} + 1 \leq (\boldsymbol{k}_W^i)^\top \hat{\boldsymbol{\lambda}}_W \leq \hat{p}_{\ell_i+1} - 1 \quad \textit{for all } i \in N \setminus W, \tag{4.1}$$

then $(\hat{\boldsymbol{\lambda}}_W, \mathbf{0}_{N\setminus W}) \in \mathbb{R}^n$ *together with* $\hat{\boldsymbol{p}}$ *forms an optimum solution of* $(\bar{H})$.

Proof. Note that $((\hat{\boldsymbol{\lambda}}_W, \mathbf{0}_{N\setminus W}), \hat{\boldsymbol{p}})$ is a feasible solution of $(\bar{H})$ since $(\boldsymbol{k}^i)^\top \begin{pmatrix} \hat{\boldsymbol{\lambda}}_W \\ \mathbf{0}_{N\setminus W} \end{pmatrix} = (\boldsymbol{k}_W^i)^\top \hat{\boldsymbol{\lambda}}_W$, $(\hat{\boldsymbol{\lambda}}_W, \hat{\boldsymbol{p}})$ is feasible to $(\bar{H}(W))$ and satisfies (4.1).

For an optimum solution $(\boldsymbol{\lambda}^*, \boldsymbol{p}^*)$ of $(\bar{H})$ let $\boldsymbol{w}^* = X\boldsymbol{\lambda}^*$, $\boldsymbol{w}_1$ be its orthogonal projection onto the range space of X_W and $\boldsymbol{w}_2 = \boldsymbol{w}^* - \boldsymbol{w}_1$. Then $\boldsymbol{w}_1 = X_W \boldsymbol{\mu}_W^*$ for some $\boldsymbol{\mu}_W^* \in \mathbb{R}^{|W|}$ and

$$(\boldsymbol{\lambda}^*)^\top K \boldsymbol{\lambda}^* = \|\boldsymbol{w}^*\|^2 \geq \|\boldsymbol{w}_1\|^2 = (\boldsymbol{\mu}_W^*)^\top K_W \boldsymbol{\mu}_W^* \tag{4.2}$$

by the orthogonality between $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. For $i \in N \cap W$ it holds that

$$\begin{aligned}(\boldsymbol{k}_W^i)^\top \boldsymbol{\mu}_W^* &= (\boldsymbol{x}^i)^\top X_W \boldsymbol{\mu}_W^* = (\boldsymbol{x}^i)^\top \boldsymbol{w}_1 = (\boldsymbol{x}^i)^\top (\boldsymbol{w}_1 + \boldsymbol{w}_2) \\ &= (\boldsymbol{x}^i)^\top \boldsymbol{w}^* = (\boldsymbol{x}^i)^\top X \boldsymbol{\lambda}^* = (\boldsymbol{k}^i)^\top \boldsymbol{\lambda}^*,\end{aligned}$$

which is between $p_{\ell_i}^* + 1$ and $p_{\ell_i+1}^* - 1$ since $(\boldsymbol{\lambda}^*, \boldsymbol{p}^*)$ is feasible to $(\bar{H})$. Then $(\boldsymbol{\mu}_W^*, \boldsymbol{p}^*)$ is feasible to $(\bar{H}(W))$. This and the optimality of $\hat{\boldsymbol{\lambda}}_W$ yield the inequality

$$(\boldsymbol{\mu}_W^*)^\top K_W \boldsymbol{\mu}_W^* \geq \hat{\boldsymbol{\lambda}}_W^\top K_W \hat{\boldsymbol{\lambda}}_W = \begin{pmatrix} \hat{\boldsymbol{\lambda}}_W \\ \mathbf{0}_{N\setminus W} \end{pmatrix}^\top K \begin{pmatrix} \hat{\boldsymbol{\lambda}}_W \\ \mathbf{0}_{N\setminus W} \end{pmatrix}. \tag{4.3}$$

The two inequalities (4.2) and (4.3) prove the optimality of $((\hat{\boldsymbol{\lambda}}_W, \mathbf{0}_{N\setminus W}), \hat{\boldsymbol{p}})$. □

Theorem 4.1. *The Algorithm RC*$\bar{H}$ *solves problem* $(\bar{H})$.

5. Kernel Technique for Hard Margin Problems

The matrix K in the primal hard margin problem $(\bar{H})$ with the dual representation of the normal vector is composed of the inner products $(\boldsymbol{x}^i)^\top \boldsymbol{x}^j$ for $i, j \in N$. This enables us to apply the *kernel technique* simply by replacing them by $\kappa(\boldsymbol{x}^i, \boldsymbol{x}^j)$ for some appropriate kernel function κ.

Let $\phi\colon \mathbb{R}^m \to \mathbb{F}$ be a function, possibly unknown, from $\mathbb{R}^m$ to some higher dimensional inner product space $\mathbb{F}$, so-called the *feature space* such that

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle$$

holds for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$, where $\langle \cdot, \cdot \rangle$ is the inner product defined on $\mathbb{F}$. In the sequel we denote $\tilde{\boldsymbol{x}} = \phi(\boldsymbol{x})$. The kernel technique considers the vectors $\tilde{\boldsymbol{x}}^i \in \mathbb{F}$ instead of $\boldsymbol{x}^i \in \mathbb{R}^m$, and finds a normal vector $\tilde{\boldsymbol{w}} \in \mathbb{F}$ and thresholds $p_1, \ldots, p_l$. Therefore the matrices X and K should be replaced by $\tilde{X}$ composed of vectors $\tilde{\boldsymbol{x}}^i$ and $\tilde{K} = \left[\langle \tilde{\boldsymbol{x}}^i, \tilde{\boldsymbol{x}}^j \rangle\right]_{i,j\in N}$, respectively. Note that the latter matrix is given as

$$\tilde{K} = \begin{bmatrix} & & \\ & \kappa(\boldsymbol{x}^i, \boldsymbol{x}^j) & \\ & & \end{bmatrix}_{i,j\in N} \tag{5.1}$$

by the kernel function κ. Denote the ith row of the matrix $\tilde{K}$ by $(\tilde{\boldsymbol{k}}^i)^\top$, then the problem to solve is

$$(\tilde{H})\quad\left|\begin{array}{ll}\text{minimize} & \boldsymbol{\lambda}^\top \tilde{K}\boldsymbol{\lambda}\\ \text{subject to} & p_{\ell_i}+1 \le (\tilde{\boldsymbol{k}}^i)^\top\boldsymbol{\lambda} \le p_{\ell_i+1}-1 \qquad \text{for } i \in N.\end{array}\right.$$

Solving $(\tilde{H})$ to find $\boldsymbol{\lambda}^*$, an optimal normal vector $\tilde{\boldsymbol{w}}^* \in \mathbb{F}$ would be given as

$$\tilde{\boldsymbol{w}}^* = \sum_{i\in N}\lambda_i^*\tilde{\boldsymbol{x}}^i,$$

which is not available in general due to the absence of an explicit representation of $\tilde{\boldsymbol{x}}^i$'s. However, the value of $\langle\tilde{\boldsymbol{w}}^*,\tilde{\boldsymbol{x}}\rangle$ can be computed for the attribute vector $\boldsymbol{x}\in\mathbb{R}^n$ of a newly-arrived object in the following way:

$$\langle\tilde{\boldsymbol{w}}^*,\tilde{\boldsymbol{x}}\rangle = \langle\sum_{i\in N}\lambda_i^*\tilde{\boldsymbol{x}}^i,\tilde{\boldsymbol{x}}\rangle = \sum_{i\in N}\lambda_i^*\langle\tilde{\boldsymbol{x}}^i,\tilde{\boldsymbol{x}}\rangle = \sum_{i\in N}\lambda_i^*\langle\phi(\boldsymbol{x}^i),\phi(\boldsymbol{x})\rangle = \sum_{i\in N}\lambda_i^*\kappa(\boldsymbol{x}^i,\boldsymbol{x}).$$

Then by locating the threshold interval determined by $\boldsymbol{p}^*$ into which this value falls, we can assign a label to the new object.

In the same way as for the hard margin problem $(\bar{H})$ we consider the sub-problem

$$(\tilde{H}(W))\quad\left|\begin{array}{ll}\text{minimize} & \boldsymbol{\lambda}_W^\top \tilde{K}_W\boldsymbol{\lambda}_W\\ \text{subject to} & p_{\ell_i}+1 \le (\tilde{\boldsymbol{k}}^i_W)^\top\boldsymbol{\lambda}_W \le p_{\ell_i+1}-1 \quad \text{for } i \in W,\end{array}\right.$$

where $\tilde{K}_W$ is the sub-matrix consisting of the rows and columns of $\tilde{K}$ with indices in W, and $(\tilde{\boldsymbol{k}}^i_W)^\top$ is the row vector of $\kappa(\boldsymbol{x}^i,\boldsymbol{x}^j)$ for $j\in W$.

Algorithm RC$\tilde{\text{H}}$ (Row and Column Generation Algorithm for $(\tilde{H})$)

```
Step 1 : Let $W^0$ be an initial working set, and let $\nu = 0$.
Step 2 : Solve $(\tilde{H}(W^\nu))$ to obtain $\boldsymbol{\lambda}_{W^\nu}$ and $\boldsymbol{p}^\nu$.
Step 3 : Let $\Delta = \{\, i \in N \setminus W^\nu \mid (\boldsymbol{\lambda}_{W^\nu},\boldsymbol{p}^\nu)$ violates $p_{\ell_i}+1 \le (\tilde{\boldsymbol{k}}^i_{W^\nu})^\top\boldsymbol{\lambda}_W \le p_{\ell_i+1}-1\,\}$.
Step 4 : If $\Delta = \emptyset$, terminate.
Step 5 : Otherwise choose $\Delta^\nu \subseteq \Delta$, let $W^{\nu+1} = W^\nu \cup \Delta^\nu$, increment $\nu$ by 1 and go to Step 2.
```

The validity of the algorithm is straightforward from the following lemma, which is proved in exactly the same way as Lemma 4.1

Lemma 5.1. *Let $(\hat{\boldsymbol{\lambda}}_W,\hat{\boldsymbol{p}})\in\mathbb{R}^{|W|+l}$ be an optimum solution of $(\tilde{H}(W))$. If*

$$\hat{p}_{\ell_i}+1 \le (\tilde{\boldsymbol{k}}^i_W)^\top\hat{\boldsymbol{\lambda}}_W \le \hat{p}_{\ell_i+1}-1 \quad \textit{for all } i\in N\setminus W,$$

then $(\hat{\boldsymbol{\lambda}}_W,\boldsymbol{0}_{N\setminus W})\in\mathbb{R}^n$ together with $\hat{\boldsymbol{p}}$ forms an optimum solution of $(\tilde{H})$.

Theorem 5.1. *The Algorithm RC$\tilde{H}$ solves problem $(\tilde{H})$.*

6. Soft Margin Problems for Non-Separable Case

6.1. Primal soft margin problem

Similarly to the binary SVM, introducing nonnegative slack variables ξ_{-i} and ξ_{+i} for $i\in N$ relaxes the hard margin constraints to the *soft margin* constraints:

$$p_{\ell_i}+1-\xi_{-i} \le \boldsymbol{w}^\top\boldsymbol{x}^i \le p_{\ell_i+1}-1+\xi_{+i} \quad \text{for } i\in N.$$

Positive values of variables ξ_{-i} and ξ_{+i} mean misclassification, hence they should be as small as possible. If we penalize positive ξ_{-i} and ξ_{+i} by adding $\sum_{i\in N}(\xi_{-i}+\xi_{+i})$ to the objective function, we have the following *primal soft margin problem.*

$$(S)\quad \left|\begin{array}{ll} \text{minimize} & \|\boldsymbol{w}\|^2 + c\,\mathbf{1}_n^\top(\boldsymbol{\xi}_- + \boldsymbol{\xi}_+) \\ \text{subject to} & p_{\ell_i} + 1 - \xi_{-i} \le (\boldsymbol{x}^i)^\top \boldsymbol{w} \le p_{\ell_i+1} - 1 + \xi_{+i} \quad \text{for } i \in N \\ & \boldsymbol{\xi}_-, \boldsymbol{\xi}_+ \ge \mathbf{0}_n, \end{array}\right.$$

where $\boldsymbol{\xi}_- = (\xi_{-1},\dots,\xi_{-n}), \boldsymbol{\xi}_+ = (\xi_{+1},\dots,\xi_{+n})$ and c is a *penalty parameter.*

Naturally, we could add the constraints

$$p_{k'} + 1 - \xi_{-i} \le (\boldsymbol{x}^i)^\top \boldsymbol{w} \le p_{k''} - 1 + \xi_{+i} \quad \text{for } k', k'' \in L \text{ such that } k' \le \ell_i < k'' \text{ for } i \in N$$

to the above formulation. It would, however, inflate the problem size and most of those constraints would be likely redundant. Therefore we will not discuss this formulation.

6.2. Dual representation

Obviously we can replace $\|\boldsymbol{w}\|^2$ and $(\boldsymbol{x}^i)^\top\boldsymbol{w}$ in the primal problem given in the preceding subsection by $\boldsymbol{\lambda}^\top K\boldsymbol{\lambda}$ and $(\boldsymbol{k}^i)^\top\boldsymbol{\lambda}$, respectively to obtain the primal problem with the dual representation of the normal vector. Then we obtain

$$(\bar{S})\quad \left|\begin{array}{ll} \text{minimize} & \boldsymbol{\lambda}^\top K\boldsymbol{\lambda} + c\,\mathbf{1}_n^\top(\boldsymbol{\xi}_- + \boldsymbol{\xi}_+) \\ \text{subject to} & p_{\ell_i} + 1 - \xi_{-i} \le (\boldsymbol{k}^i)^\top \boldsymbol{\lambda} \le p_{\ell_i+1} - 1 + \xi_{+i} \quad \text{for } i \in N \\ & \boldsymbol{\xi}_-, \boldsymbol{\xi}_+ \ge \mathbf{0}_n. \end{array}\right.$$

7. Algorithms for Soft Margin Problems

The algorithms for the soft margin problems may not differ substantially from those for the hard margin problems. The relaxed problem $(S(W))$ of (S) for the working set $W \subseteq N$ is

$$(S(W))\quad \left|\begin{array}{ll} \text{minimize} & \|\boldsymbol{w}\|^2 + c\,\mathbf{1}_{|W|}^\top(\boldsymbol{\xi}_{-W} + \boldsymbol{\xi}_{+W}) \\ \text{subject to} & p_{\ell_i} + 1 - \xi_{-i} \le (\boldsymbol{x}^i)^\top \boldsymbol{w} \le p_{\ell_i+1} - 1 + \xi_{+i} \quad \text{for } i \in W \\ & \boldsymbol{\xi}_{-W}, \boldsymbol{\xi}_{+W} \ge \mathbf{0}_{|W|}, \end{array}\right.$$

where $\boldsymbol{\xi}_{-W} = (\xi_{-i})_{i\in W}$ and $\boldsymbol{\xi}_{+W} = (\xi_{+i})_{i\in W}$.

Lemma 7.1. *If an optimum solution* $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{\xi}}_{-W}, \hat{\boldsymbol{\xi}}_{+W})$ *of problem* $(S(W))$ *satisfies the constraints*

$$\hat{p}_{\ell_i} + 1 \le (\boldsymbol{x}^i)^\top \hat{\boldsymbol{w}} \le \hat{p}_{\ell_i+1} - 1 \quad \textit{for all } i \in N \setminus W,$$

then $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{p}}, (\hat{\boldsymbol{\xi}}_{-W}, \mathbf{0}_{N\setminus W}), (\hat{\boldsymbol{\xi}}_{+W}, \mathbf{0}_{N\setminus W}))$ *is an optimum solution of* (S).

Proof. Clearly $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{\xi}_-}, \hat{\boldsymbol{\xi}}_+) = (\hat{\boldsymbol{w}}, \hat{\boldsymbol{p}}, (\hat{\boldsymbol{\xi}}_{-W}, \mathbf{0}_{N\setminus W}), (\hat{\boldsymbol{\xi}}_{+W}, \mathbf{0}_{N\setminus W}))$ is a feasible solution of (S). To show its optimality, we suppose that there is another feasible solution $(\boldsymbol{w}, \boldsymbol{p}, \boldsymbol{\xi}_-, \boldsymbol{\xi}_+)$ of (S) such that

$$\|\hat{\boldsymbol{w}}\|^2 + c\,\mathbf{1}_n^\top(\hat{\boldsymbol{\xi}}_- + \hat{\boldsymbol{\xi}}_+) > \|\boldsymbol{w}\|^2 + c\,\mathbf{1}_n^\top(\boldsymbol{\xi}_- + \boldsymbol{\xi}_+).$$

Since $\hat{\boldsymbol{\xi}}_{-N\setminus W} = \hat{\boldsymbol{\xi}}_{+N\setminus W} = \mathbf{0}_{N\setminus W}$ and $\boldsymbol{\xi}_{-N\setminus W}, \boldsymbol{\xi}_{+N\setminus W} \ge \mathbf{0}_{N\setminus W}$ we obtain the inequality

$$\|\hat{\boldsymbol{w}}\|^2 + c\,\mathbf{1}_{|W|}^\top(\hat{\boldsymbol{\xi}}_{-W} + \hat{\boldsymbol{\xi}}_{+W}) > \|\boldsymbol{w}\|^2 + c\,\mathbf{1}_{|W|}^\top(\boldsymbol{\xi}_{-W} + \boldsymbol{\xi}_{+W}).$$

This contradicts the optimality of $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{\xi}}_{-W}, \hat{\boldsymbol{\xi}}_{+W})$ since $(\boldsymbol{w}, \boldsymbol{p}, \boldsymbol{\xi}_{-W}, \boldsymbol{\xi}_{+W})$ is feasible to $(S(W))$. □

Therefore the following algorithm will solve problem (S) when it terminates.

Algorithm RS (Row Generation Algorithm for (S))

```
Step 1 : Let $W^0$ be an initial working set and let $\nu = 0$.
Step 2 : Solve $(S(W^\nu))$ to obtain $(\boldsymbol{w}^\nu, \boldsymbol{p}^\nu, \boldsymbol{\xi}_{-W^\nu}, \boldsymbol{\xi}_{+W^\nu})$.
Step 3 : Let $\Delta = \{\, i \in N \setminus W^\nu \mid (\boldsymbol{w}^\nu, \boldsymbol{p}^\nu)$ violates $p_{\ell_i} + 1 \le (\boldsymbol{x}^i)^\top \boldsymbol{w} \le p_{\ell_i+1} - 1 \,\}$.
Step 4 : If $\Delta = \emptyset$, terminate.
Step 5 : Otherwise choose $\Delta^\nu \subseteq \Delta$, let $W^{\nu+1} = W^\nu \cup \Delta^\nu$, increment $\nu$ by 1 and go to Step 2.
```

Next we consider the primal soft margin problem $(\bar{S})$ with the dual representation of the normal vector. The sub-problem to solve is

$$(\bar{S}(W)) \quad \left| \begin{array}{ll} \text{minimize} & \boldsymbol{\lambda}_W^\top K_W \boldsymbol{\lambda}_W + c\,\mathbf{1}_{|W|}^\top(\boldsymbol{\xi}_{-W} + \boldsymbol{\xi}_{+W}) \\ \text{subject to} & p_{\ell_i} + 1 - \xi_{-i} \le (\boldsymbol{k}_W^i)^\top \boldsymbol{\lambda}_W \le p_{\ell_i+1} - 1 + \xi_{+i} \quad \text{for } i \in W \\ & \boldsymbol{\xi}_{-W},\ \boldsymbol{\xi}_{+W} \ge \mathbf{0}_{|W|}. \end{array} \right.$$

Algorithm RC$\bar{\mathbf{S}}$ (Row and Column Generation Algorithm for $(\bar{S})$)

```
Step 1 : Let $W^0$ be an initial working set, and let $\nu = 0$.
Step 2 : Solve $(\bar{S}(W^\nu))$ to obtain $(\boldsymbol{\lambda}_{W^\nu}, \boldsymbol{p}^\nu, \boldsymbol{\xi}_{-W^\nu}, \boldsymbol{\xi}_{+W^\nu})$.
Step 3 : Let $\Delta = \{\, i \in N \setminus W^\nu \mid (\boldsymbol{\lambda}_{W^\nu}, \boldsymbol{p}^\nu)$ violates $p_{\ell_i} + 1 \le (\boldsymbol{k}_{W^\nu}^i)^\top \boldsymbol{\lambda}_W \le p_{\ell_i+1} - 1 \,\}$.
Step 4 : If $\Delta = \emptyset$, terminate.
Step 5 : Otherwise choose $\Delta^\nu \subseteq \Delta$, let $W^{\nu+1} = W^\nu \cup \Delta^\nu$, increment $\nu$ by 1 and go to Step 2.
```

Lemma 7.2. *Let* $(\hat{\boldsymbol{\lambda}}_W, \hat{\boldsymbol{p}}, \hat{\boldsymbol{\xi}}_{-W}, \hat{\boldsymbol{\xi}}_{+W})$ *be an optimum solution of* $(\bar{S}(W))$. *If*

$$\hat{p}_{\ell_i} + 1 \le (\boldsymbol{k}_W^i)^\top \hat{\boldsymbol{\lambda}}_W \le \hat{p}_{\ell_i+1} - 1 \quad \textit{for all } i \in N \setminus W,$$

then $((\hat{\boldsymbol{\lambda}}_W, \mathbf{0}_{N\setminus W}), \hat{\boldsymbol{p}}, (\boldsymbol{\xi}^\nu_{-W}, \mathbf{0}_{N\setminus W}), (\boldsymbol{\xi}^\nu_{+W}, \mathbf{0}_{N\setminus W}))$ *is an optimum solution of* $(\bar{S})$.

Proof. First note that $((\hat{\boldsymbol{\lambda}}_W, \mathbf{0}_{N\setminus W}), \hat{\boldsymbol{p}}, (\hat{\boldsymbol{\xi}}_{-W}, \mathbf{0}_{N\setminus W}), (\hat{\boldsymbol{\xi}}_{+W}, \mathbf{0}_{N\setminus W}))$ is feasible to $(\bar{S})$. Let $(\boldsymbol{\lambda}^*, \boldsymbol{p}^*, \boldsymbol{\xi}^*_{-W}, \boldsymbol{\xi}^*_{+W})$ be an optimum solution of $(\bar{S})$, let $\boldsymbol{w}^* = X\boldsymbol{\lambda}^*$ and $\boldsymbol{w}_1$ be its orthogonal projection onto the range space of X_W. Then we see that the coefficient vector $\boldsymbol{\mu}^*_W$ such that $\boldsymbol{w}_1 = X_W \boldsymbol{\mu}^*_W$ together with $(\boldsymbol{p}^*, \boldsymbol{\xi}^*_{-W}, \boldsymbol{\xi}^*_{+W})$ forms a feasible solution of $(\bar{S}(W))$, and $\|\boldsymbol{w}^*\| \ge \|\boldsymbol{w}_1\|$. Therefore in the same manner as Lemma 4.1, we obtain the desired result. □

The validity of the algorithm directly follows the above lemma.

Theorem 7.1. *The Algorithm RC$\bar{S}$ solves problem* $(\bar{S})$.

8. Kernel Technique for Soft Margin Problems

The kernel technique can apply to the soft margin problem in the same way as discussed in Section 5.

For the kernel version of soft margin problems with the dual representation of the normal vector, we have only to replace K by $\tilde{K}$ given by some kernel function κ. Then the kernel version of $(\bar{S})$ is given as

$$
(\tilde{S}) \quad \left|\begin{array}{ll}
\text{minimize} & \boldsymbol{\lambda}^\top \tilde{K} \boldsymbol{\lambda} + c\,\mathbf{1}_n^\top(\boldsymbol{\xi}_- + \boldsymbol{\xi}_+) \\
\text{subject to} & p_{\ell_i} + 1 - \xi_{-i} \leq (\tilde{\boldsymbol{k}}^i)^\top \boldsymbol{\lambda} \leq p_{\ell_i+1} - 1 + \xi_{+i} \quad \text{for } i \in N \\
& \boldsymbol{\xi}_-, \boldsymbol{\xi}_+ \geq \mathbf{0}_n.
\end{array}\right.
$$

In the same way as in the previous section we consider the sub-problem of $(\tilde{S})$, which is given as

$$
(\tilde{S}(W)) \quad \left|\begin{array}{ll}
\text{minimize} & \boldsymbol{\lambda}_W^\top \tilde{K}_W \boldsymbol{\lambda}_W + c\,\mathbf{1}_{|W|}^\top(\boldsymbol{\xi}_{-W} + \boldsymbol{\xi}_{+W}) \\
\text{subject to} & p_{\ell_i} + 1 - \xi_{-i} \leq (\tilde{\boldsymbol{k}}_W^i)^\top \boldsymbol{\lambda}_W \leq p_{\ell_i+1} - 1 + \xi_{+i} \quad \text{for } i \in W \\
& \boldsymbol{\xi}_{-W},\, \boldsymbol{\xi}_{+W} \geq \mathbf{0}_{|W|}.
\end{array}\right.
$$

```
Algorithm RC$\tilde{\mathbf{S}}$ (Row and Column Generation Algorithm for $(\tilde{S})$)
Step 1 : Let $W^0$ be an initial working set, and let $\nu = 0$.
Step 2 : Solve $(\tilde{S}(W^\nu))$ to obtain $(\boldsymbol{\lambda}_{W^\nu}, \boldsymbol{p}^\nu, \boldsymbol{\xi}_{-W^\nu}, \boldsymbol{\xi}_{+W^\nu})$.
Step 3 : Let $\Delta = \{\, i \in N \setminus W^\nu \mid (\boldsymbol{\lambda}_{W^\nu}, \boldsymbol{p}^\nu)$ violates $p_{\ell_i} + 1 \leq (\tilde{\boldsymbol{k}}_{W^\nu}^i)^\top \boldsymbol{\lambda}_W \leq p_{\ell_i+1} - 1 \,\}$.
Step 4 : If $\Delta = \emptyset$, terminate.
Step 5 : Otherwise choose $\Delta^\nu \subseteq \Delta$, let $W^{\nu+1} = W^\nu \cup \Delta^\nu$, increment $\nu$ by 1 and go to Step 2.
```

We then obtain the following theorem.

Theorem 8.1. *The Algorithm RC$\tilde{S}$ solves problem $(\tilde{S})$.*

9. Illustrative Example

We show with a small instance how different models result in different classifications. The instance is the grades in calculus of 44 undergraduates. Each student is given one of the four possible grades A, B, C and D according to his/her total score of mid-term exam, end-of-term exam and a number of in-class quizzes. We take the scores of student's mid-term and end-of-term exams to form the attribute vector, and his/her grade as the label.

Since the score of quizzes is not considered as an attribute, the instance is not separable, hence the hard margin problem (H) is infeasible. The solution of the soft margin problem (S) with $c = 15$ is given in Figure 1, where the students of different grades are loosely separated by three straight lines. Each value on the lines in the figure represents the corresponding threshold p_k.

Using the following two kernel functions defined as

$$
\begin{aligned}
\kappa(\boldsymbol{x}, \boldsymbol{y}) &= \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{y}\|^2\right) && \text{(Gaussian kernel)}, \\
\kappa(\boldsymbol{x}, \boldsymbol{y}) &= (1 + \boldsymbol{x}^\top \boldsymbol{y})^d && \text{(Polynomial kernel)}
\end{aligned}
$$

with several different values of σ and d, we solved $(\tilde{S})$. The result of the Gaussian kernel with $c = 10$ and $\sigma = 0.5$ is given in Figure 2, where one can observe that the problem $(\tilde{S})$ with the Gaussian kernel is exposed to the risk of over-fitting. This issue will be discussed in Appendix. The result of the polynomial kernel with $c = 15$ and $d = 4$ is given in Figure 3. From Fig. 3, we observe that the students of different grades are separated by three gentle curves.

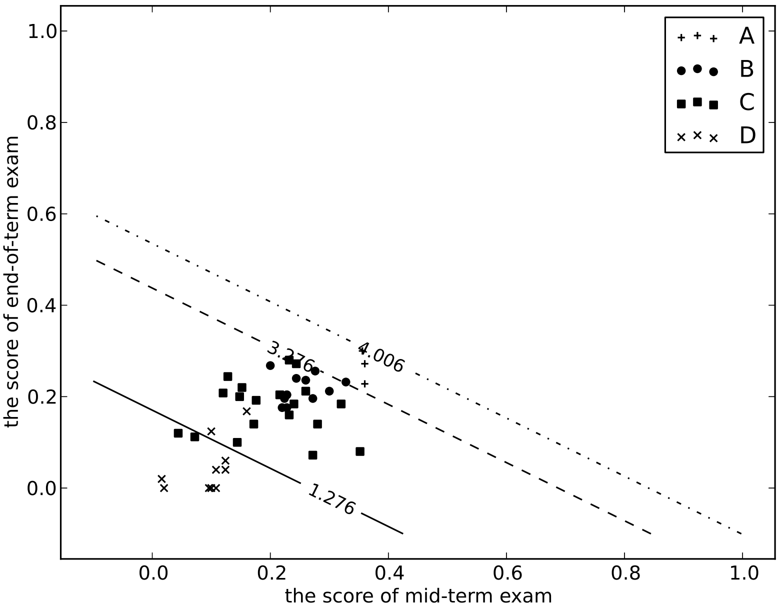

Figure 1: Classification by (S)

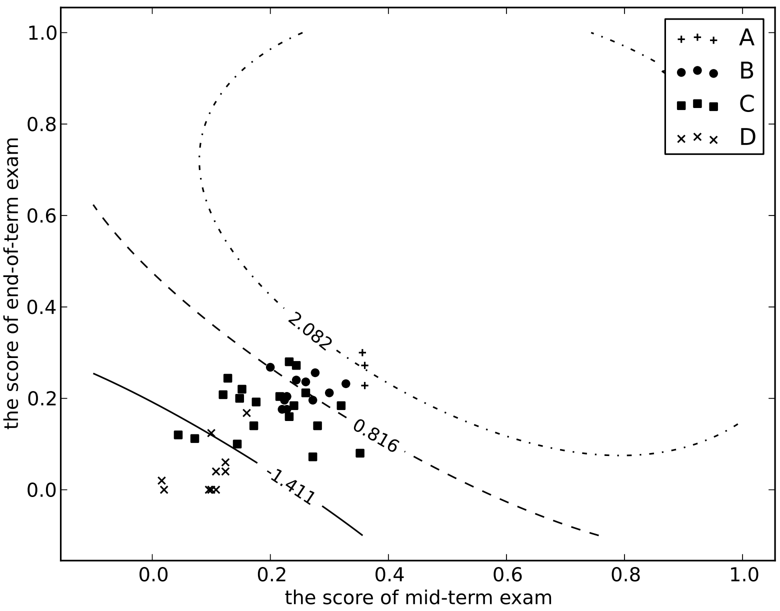

Figure 2: Classification by $(\tilde{S})$ with the Gaussian kernel

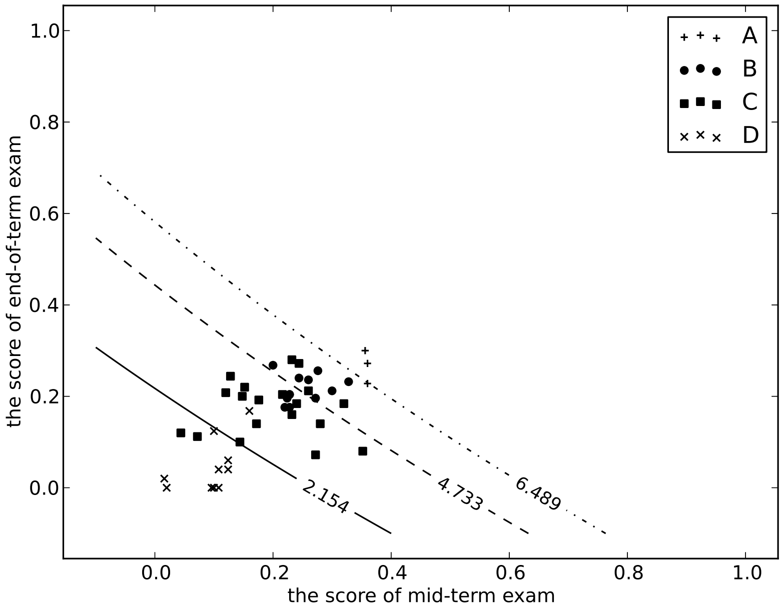

Figure 3: Classification by $(\tilde{S})$ with the polynomial kernel

10. Computational Experiments

We report on the computational experiments with our proposed algorithms. Implementing the algorithms in Python 2.7, using Gurobi 6.0.0 as a QP solver, we performed the experiments on a PC with Intel Core i7, 3.70 GHz processor and 32.0 GB of memory.

The instances we tested were randomly generated and fall into two types: separable instances and non-separable instances. First, we generate n attribute vectors $\boldsymbol{x}^i = (x_1^i, x_2^i)$ of two dimension, each component of which is drawn uniformly from the unit interval $[0, 1]$. Then object i is assigned the label ℓ_i defined as

$$\ell_i = \max_{\ell \in L} \{ \, \ell \in L \mid x_1^i + x_2^i > p_\ell \, \}$$

for the fixed thresholds $(p_0, p_1, p_2, p_3) = (-\infty, 0.5, 1.0, 1.5)$. The instances thus generated are of the first type, i.e., separable. Non-separable instances are generated by altering the labels of objects. Namely, adding a random noise to each element of the attribute vector to make a perturbed attribute vector $(x_1^i + \varepsilon_1^i, x_2^i + \varepsilon_2^i)$, where ε_1^i and ε_2^i follow the normal distribution with a zero mean and a standard deviation of 0.03, we give the label ℓ_i to object i according to the sum $x_1^i + \varepsilon_1^i + x_2^i + \varepsilon_2^i$ instead of $x_1^i + x_2^i$. Due to the presence of the random noise, the instances thus generated are not necessarily separable.

We generate five datasets for each instance with several different number of objects since the results may change owing to the random variables used in the instance generation. We name the separable type dataset (resp., the non-separable dataset) "S.n.q" (resp., "NS.n.q"), where n is the number of objects and q is the dataset ID.

We solved the separable instances by the algorithm RC$\tilde{\text{H}}$ and the non-separable instances by RC$\tilde{\text{S}}$ with $c = 10$. In all experiments we used the polynomial kernel with $d = 4$. To make the initial working set W^0, we collect three objects for each label that have the highest, the lowest and the median values of $x_1^i + x_2^i$ among the objects assigned the same label. At Step 5 in the algorithms, we add to the current working set W^ν at most two objects corresponding to the most violated constraints at $(\boldsymbol{\lambda}_{W^\nu}, \boldsymbol{p}^\nu)$, more precisely, we add the objects i and $j \in N \setminus W^\nu$ such that

$$i = \mathrm{argmax} \left\{ 1 - (\tilde{\boldsymbol{k}}^i_{W^\nu})^\top \boldsymbol{\lambda}_{W^\nu} + p^\nu_{\ell_i} > 0 \mid i \in N \setminus W^\nu \right\},$$
$$j = \mathrm{argmax} \left\{ 1 + (\tilde{\boldsymbol{k}}^i_{W^\nu})^\top \boldsymbol{\lambda}_{W^\nu} - p^\nu_{\ell_i+1} > 0 \mid i \in N \setminus W^\nu \right\}.$$

Table 1 shows the computational results of applying RC$\tilde{\text{H}}$ (resp., RC$\tilde{\text{S}}$) to separable instances (resp., non-separable instances), where the columns "# iter.", "# added obj." and "time" represent the number of sub-problems solved, the number of added objects and the computation time in seconds, respectively. In order to assess the efficiency of our algorithm, we added the column "GRB" showing the computation time when the whole problems $(\tilde{H})$ and $(\tilde{S})$ were directly input and solved by Gurobi, and the column "PRank" showing the computation time of applying PRank algorithm [2], which is an on-line learning algorithm motivated by the perceptron. The entries "ave." and "st.dev." show the average and the standard deviation across the five datasets.

We begin by looking at the results for the separable instances. From Table 1, we observe that the number of added objects is much smaller than that of the original problem. Specifically, only from 0.4% to 8.6% of the whole set of variables of the problems suffices in order to obtain an optimal solution. Thus RC$\tilde{\text{H}}$ requires a small memory capacity, and it helps RC$\tilde{\text{H}}$ be applicable to further larger instances. Gurobi Optimizer for solving $(\tilde{H})$ requires a rather long computation time as the instance size grows. To be specific, Gurobi takes over 10,000 seconds, approximately 2.8 hours, on average to solve the instances with $n = 10{,}000$, while our algorithm takes less than 300 seconds on average. Concerning PRank algorithm, it also requires a long computation time for large instances since the algorithm needs to compute and store an n-dimensional row vector of $\tilde{K}$ at every iteration. From these observations, RC$\tilde{\text{H}}$ is superior to applying Gurobi and PRank directly to the whole problem in terms of both computation time and memory consumption.

Turning now to the results for the non-separable instances, we observe that the number of iterations as well as the number of added objects is larger than that for the separable instances. Nevertheless the number of added objects is approximately 20% of the whole set of variables, that is, RC$\tilde{\text{S}}$ also requires a small memory capacity. In contrast, RC$\tilde{\text{S}}$ takes much longer computation time than applying Gurobi and PRank directly to the whole problem. This drawback is due to the way of our implementation. Whenever a working set W is updated in our algorithm, we generate a sub-problem corresponding to W by adding not only constraints but also variables as well. The vast majority of computation time was spent for Gurobi to carry out sub-problem generation repeatedly every time when the working set was updated. Take "NS.1000.5" for instance, we give a breakdown of the computation time at each iteration in Figure 4. The legends in the figure are as follows: "GenTime" is the time of generating a sub-problem, "RunTime" is the time of solving a sub-problem, and "FeasTime" is the time required for searching violated constraints at Step 3. We observe that the predominant portion of the computation time was devoted to the problem generation and the run time and feasibility checking time together are fringe. This suggests that a finely tuned application of a QP solver could dramatically reduce the total computation time. This would merit further research.

Next, we compare our algorithms with PRank algorithm with respect to the accuracy of

Table 1: Computational results

instance	n	# iter.		# added obj.		time (sec.)		GRB (sec.)		PRank (sec.)	
		ave.	st.dev.	ave.	st.dev.	ave.	st.dev.	ave.	st.dev.	ave.	st.dev.
S.100.1 – S.100.5	100	5.800	1.789	8.600	3.050	0.336	0.174	0.004	0.000	0.028	0.001
S.500.1 – S.500.5	500	8.200	1.643	13.600	2.702	1.018	0.313	0.099	0.006	0.562	0.032
S.1000.1 – S.1000.5	1000	12.200	0.837	20.800	1.095	3.342	0.353	0.499	0.015	2.644	0.259
S.5000.1 – S.5000.5	5000	19.400	3.975	35.600	7.301	65.724	14.942	1139.662	389.168	207.425	1.027
S.10000.1 – S.10000.5	10000	23.400	6.189	42.600	11.349	286.731	80.009	10639.340	3839.160	1472.864	2.497
NS.100.1 – NS.100.5	100	16.400	7.403	25.200	6.058	2.592	1.859	0.022	0.002	0.027	0.002
NS.500.1 – NS.500.5	500	65.600	31.777	109.800	24.448	107.200	106.311	1.601	0.066	0.544	0.004
NS.1000.1 – NS.1000.5	1000	108.400	39.855	193.600	36.184	493.781	444.807	15.470	4.984	2.462	0.018

Table 2: Average rank losses

instance	n	Our algorithm		PRank	
		ave.	st.dev.	ave.	st.dev.
S.100.1 – S.100.5	100	0.000	0.000	0.458	0.390
S.500.1 – S.500.5	500	0.000	0.000	0.250	0.166
S.1000.1 – S.1000.5	1000	0.000	0.000	0.244	0.180
S.5000.1 – S.5000.5	5000	0.000	0.000	0.101	0.045
S.10000.1 – S.10000.5	10000	0.000	0.000	0.065	0.023
NS.100.1 – NS.100.5	100	0.058	0.015	0.292	0.060
NS.500.1 – NS.500.5	500	0.059	0.009	0.192	0.088
NS.1000.1 – NS.1000.5	1000	0.063	0.007	0.260	0.255

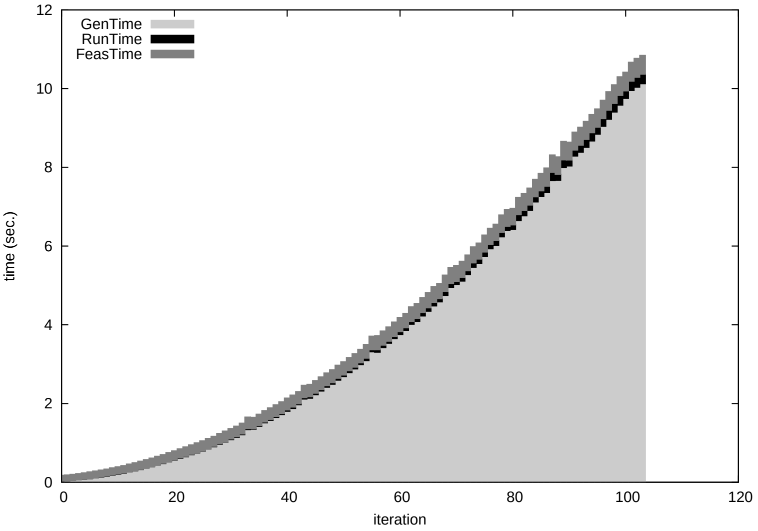

Figure 4: Breakdown of the computation time for "NS.1000.5"

solutions. We use the following average rank loss as a measure of accuracy:

$$\frac{1}{n}\sum_{i\in N}|\hat{\ell}_i - \ell_i|,$$

where $\hat{\ell}_i$ is a label predicted from the solution $(\boldsymbol{\lambda}^*, \boldsymbol{p}^*)$ obtained by the algorithms. Namely, $\hat{\ell}_i = \max_{\ell\in L}\{\,\ell \in L \mid \sum_{j\in N}\lambda^*_j\kappa(\boldsymbol{x}^i, \boldsymbol{x}^j) > p^*_\ell\,\}$. Since a positive value of the average rank loss indicates misclassification, it is preferable to be close to zero. Table 2 summarizing the average rank losses of our algorithms and PRank algorithm shows that our algorithms outperform PRank algorithm in all instances. As a matter of course, we confirm that the average rank losses obtained by our algorithm are zero for the separable instances.

11. Conclusions

In this paper, we proposed to apply the dual representation of the normal vector to the formulation based on the fixed margin strategy by Shashua and Levin [6] for the ranking problem. The problem obtained has the drawback that it has n of variables as well as n of constraints. However the fact that it enables an application of the kernel technique outweighs the drawback. Then we proposed a row and column generation algorithm. Namely, we start the algorithm with a sub-problem which is much smaller than the master problem in both variables and constraints, and increment both of them as the computation goes on. Furthermore we proved the validity of the algorithm. Through some preliminary experiments, our algorithm performed fairly well. However it should need further research such as the setting of the initial working set W^0 and the choice of Δ^ν since a clever choice of these may enhance the efficiency of the algorithm.

Acknowledgements

The authors would like to thank the referees for their careful review and valuable comments. The last author is partly supported by The Okawa Foundation for Information and Telecommunications.

Appendix A. Monotonicity Issue

In some situations it would be desirable that the separating curves have some monotonicity property, namely an object with attribute vector $\boldsymbol{x}$ be ranked higher than an object with $\boldsymbol{y}$ such that $\boldsymbol{y} \leq \boldsymbol{x}$.

Let P be a hyperplane in $\mathbb{F}$ defined by

$$P = \{\, \tilde{\boldsymbol{x}} \in \mathbb{F} \mid \langle \tilde{\boldsymbol{w}}^*, \tilde{\boldsymbol{x}} \rangle = b \,\}$$

for some constant $b \in \mathbb{R}$ and let C denote its inverse image under the unknown function ϕ, i.e.,

$$C = \{\, \boldsymbol{x} \in \mathbb{R}^m \mid \phi(\boldsymbol{x}) \in P \,\}.$$

Then $\boldsymbol{x} \in C$ if and only if $\langle \tilde{\boldsymbol{w}}^*, \phi(\boldsymbol{x}) \rangle = b$. Since $\tilde{\boldsymbol{w}}^* = \sum_{i\in N} \lambda_i^* \tilde{\boldsymbol{x}}^i = \sum_{i\in N} \lambda_i^* \phi(\boldsymbol{x}^i)$, we obtain

$$\langle \sum_{i\in N} \lambda_i^* \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}) \rangle = b.$$

Due to the bi-linearity of the inner product $\langle \cdot, \cdot \rangle$, we have

$$\langle \sum_{i\in N} \lambda_i^* \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}) \rangle = \sum_{i\in N} \lambda_i^* \langle \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}) \rangle = \sum_{i\in N} \lambda_i^* \kappa(\boldsymbol{x}^i, \boldsymbol{x}),$$

and then an expression of the inverse image

$$C = \{\, \boldsymbol{x} \in \mathbb{R}^m \mid \sum_{i\in N} \lambda_i^* \kappa(\boldsymbol{x}^i, \boldsymbol{x}) = b \,\}$$

by the kernel function κ.

Suppose that the kernel function $\kappa(\boldsymbol{x}^i, \cdot)$ is nondecreasing for $i \in N$, in the sense that

$$\boldsymbol{x} \leq \boldsymbol{x}' \Rightarrow \kappa(\boldsymbol{x}^i, \boldsymbol{x}) \leq \kappa(\boldsymbol{x}^i, \boldsymbol{x}'),$$

and $\lambda_i^* \geq 0$ for $i \in N$. Then $\sum_{i\in N} \lambda_i^* \kappa(\boldsymbol{x}^i, \boldsymbol{x})$ is nondecreasing as a whole.

Lemma A.1. *The kernel function $\kappa(\boldsymbol{x}^i, \cdot)$ is nondecreasing and $\lambda_i^* \geq 0$ for $i \in N$. Then the contours are nondecreasing.*

The polynomial kernel

$$\kappa(\boldsymbol{x}^i, \boldsymbol{x}) = (1 + (\boldsymbol{x}^i)^\top \boldsymbol{x})^d$$

is nondecreasing with respect to $\boldsymbol{x}$ if $\boldsymbol{x}^i \geq \mathbf{0}$ for $i \in N$. Therefore it would be appropriate to use the polynomial kernel when all the attribute vectors are nonnegative including those of potential objects, and the monotonicity is desirable. In this case the kernel hard margin problem $(\tilde{H})$ should be added non-negativity constraints of variables λ_i's:

$$(\tilde{H}_+) \quad \left| \begin{array}{lll} \text{minimize} & \boldsymbol{\lambda}^\top \tilde{K} \boldsymbol{\lambda} & \\ \text{subject to} & p_{\ell_i} + 1 \leq (\tilde{\boldsymbol{k}}^i)^\top \boldsymbol{\lambda} \leq p_{\ell_i + 1} - 1 & \text{for } i \in N \\ & \boldsymbol{\lambda} \geq \mathbf{0}. & \end{array} \right.$$

The problem remains an ordinary convex quadratic optimization, and the additional non-negativity constraints do not make it more difficult to solve.

References

[1] C.M. Bishop: *Pattern Recognition and Machine Learning* (Springer, New York, 2006).

[2] K. Crammer and Y. Singer: Pranking with ranking. In T.G. Dietterich, S. Becker, and Z. Ghahramani (eds.): *Advances in Neural Information Processing Systems 14 (NIPS 2001)* (MIT Press, Cambridge, 2002), 641–647.

[3] R. Herbrich, T. Graepel, and K. Obermayer: Large margin rank boundaries for ordinal regression. In A.J. Smola, P. Bartlette, B. Schölkopt, and D. Schuurmans (eds.): *Advances in Large Margin Classifiers* (MIT Press, Cambridge, 2000), 115–132.

[4] T.-Y. Liu: *Learning to Rank for Information Retrieval* (Springer-Verlag, Heidelberg, 2011).

[5] B. Schölkopf, R. Herbrich, and A.J. Smola: A generalized representer theorem. In D. Helmbold and B. Williamson (eds.): *Computational Learning Theory, Lecture Notes in Computer Science*, **2111** (2001), 416–426.

[6] A. Shashua and A. Levin: Ranking with large margin principles: two approaches. In S. Becker, S. Thrun and K. Obermayer (eds.): *Advances in Neural Information Processing Systems 15 (NIPS 2002)* (MIT Press, Cambridge, 2003), 937–944.

[7] J. Shawe-Taylor and N. Cristianini: *Kernel Methods for Pattern Analysis* (Cambridge University Press, Cambridge, 2004).

[8] K. Tatsumi, K. Hayashida, R. Kawachi, and T. Tanino: Multiobjective multiclass support vector machines maximizing geometric margins. *Pacific Journal of Optimization*, **6** (2010), 115–140.

[9] V.N. Vapnik: *Statistical Learning Theory* (John-Wiley & Sons, New York, 1998).

Yoichi Izunaga
University of Tsukuba
1-1-1 Tennodai, Tsukuba
Ibaraki 305-8573 Japan
E-mail: `s1130131@sk.tsukuba.ac.jp`