PAPER

Efficient Reformulation of 1-Norm Ranking SVM*

Daiki SUEHIRO[†,††a)], **Kohei HATANO**[††,†††], ***and*** **Eiji TAKIMOTO**[††††], ***Members***

SUMMARY Finding linear functions that maximize AUC scores is important in ranking research. A typical approach to the ranking problem is to reduce it to a binary classification problem over a new instance space, consisting of all pairs of positive and negative instances. Specifically, this approach is formulated as hard or soft margin optimization problems over pn pairs of p positive and n negative instances. Solving the optimization problems directly is impractical since we have to deal with a sample of size pn, which is quadratically larger than the original sample size $p + n$. In this paper, we reformulate the ranking problem as variants of hard and soft margin optimization problems over $p+n$ instances. The resulting classifiers of our methods are guaranteed to have a certain amount of AUC scores.

key words: *bipartite ranking, AUC, Ranking SVMs*

1. Introduction

Learning to rank has been one of the most active areas of research in machine learning and information retrieval in the past decade, due to increasing demands in, for example, recommendation tasks and financial risk analysis [2], [5], [8], [9], [12], [16], [20], [26], [27]. Among the problems related to learning to rank, the bipartite ranking is a fundamental problem, which involves learning to obtain rankings over positive and negative instances [11], [12], [16], [17]. More precisely, for a given sample consisting of positive and negative instances, the goal of the bipartite ranking problem is to find a real-valued function h, which is referred to as a ranking function, with the following property: For a randomly chosen test pair of positive instance $\boldsymbol{x}^+$ and negative instance $\boldsymbol{x}^-$, the ranking function h maps $\boldsymbol{x}^+$ to a higher value than $\boldsymbol{x}^-$ with high probability. Thus, a natural measure for evaluating the goodness of ranking function h is the probability that $h(\boldsymbol{x}^+) > h(\boldsymbol{x}^-)$, which we call the AUC (the area under the receiver operating characteristic (ROC) curve) of h. Note that maximizing AUC cannot be achieved by simply using binary classification algorithms such as SVM (Support Vector Machine). Although the sign of the function values $h(\boldsymbol{x})$ output by SVM indicate positive or negative class of $h(\boldsymbol{x})$, the magnitude of the values does not indicate the level of positiveness or negativeness. In other words, SVM outputs a good classification rule by margin-based theory, but does not always outputs a good ranking rule. We show an example of difference between AUC and classification accuracy in Fig. 1. Both $h(\boldsymbol{x})$ of (a) and (b) achieves same classification accuracy, however, (b) achieves higher AUC than (a). Because in (a), the value of the left (+) is less than right two (−)s, but in (b), the value of the left (+) is less than only one (−). We can see that maximizing AUC is different from maximizing classification accuracy. Therefore, when given binary labeled data, if we want to obtain a function which achieves good AUC, bipartite ranking is reasonable problem setting.

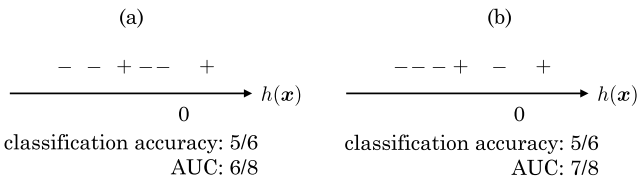

Fig. 1 AUC and classification accuracy.

It is known that the bipartite ranking problem can be reduced to a binary classification problem over a new instance space, consisting of all pairs $(\boldsymbol{x}^+, \boldsymbol{x}^-)$ of positive and negative instances. More precisely, the problem of maximizing the AUC is equivalent to finding a binary classifier f of the form of $f(\boldsymbol{x}^+, \boldsymbol{x}^-) = h(\boldsymbol{x}^+) - h(\boldsymbol{x}^-)$ so that the probability that $f(\boldsymbol{x}^+, \boldsymbol{x}^-) > 0$ is maximized for a randomly chosen instance pair. Several studies including the Ranking SVM have taken this approach with a linear classifier $f(\boldsymbol{x}^+, \boldsymbol{x}^-) = \boldsymbol{w} \cdot (\boldsymbol{x}^+ - \boldsymbol{x}^-)$ for some weight vector $\boldsymbol{w}$ as the ranking function. The Ranking SVM is justified by generalization bounds [21] which say that a large margin over pairs of positive and negative instances in the sample implies a high AUC score under the standard assumption that instances are drawn i.i.d. under the underlying distribution.

Note that a naive implementation of the reduction approach for bipartite ranking problem is impractical since the sample constructed through the reduction (called the pair-sample) is of size pn when the original sample consists of p positive and n negative instances. This implies

Manuscript received July 18, 2017.
Manuscript revised November 21, 2017.
Manuscript publicized December 4, 2017.

[†]The author is with Department of Advanced Information Technology, Information Science and Electrical Engineering, Kyushu University, Fukuoka-shi, 819–0395 Japan.
[††]The authors are with AIP, RIKEN, Tokyo, 103–0027 Japan.
[†††]The author is with Faculty of Arts and Science, Kyushu University, Fukuoka-shi, 819–0395 Japan.
[††††]The author is with Department of Informatics, Information Science and Electrical Engineering, Kyushu University, Fukuoka-shi, 819–0395 Japan.

*The previous version of this paper has already been published [30]. However, in this paper, we add the following three important subjects, the notion of AUC consistency, supplementary experiments, and reformulation of the 2-norm Ranking SVM.

a) E-mail: suehiro@ait.kyushu-u.ac.jp
DOI: 10.1587/transinf.2017EDP7233

that we need to solve a quadratic problem (QP) problem of size $O(pn(pn+N))$ in the primal form or of size $O((pn)^2)$ in the dual form (with a kernel), where N is the dimension of the instance space. To overcome this inefficiency, Joachims proposed a Cutting-Plane based algorithm called the SVM-Perf [17], which simulates the Ranking SVM in $O(s(p+n)\log(p+n))$ time, where s is the maximum number of non-zero features in the given instances. Chapelle and Keerthi proposed another efficient implementation of the Ranking SVM, called the PRSVM [7], which runs in $O((p+n)\log(p+n)+N(p+n))$ time. Note that the Ranking SVM simulated in the above implementations is based on the standard SVM formulation. That is, they solve a soft margin optimization problem with 2-norm of the weight vector regularized.

In this paper, we consider a 1-norm SVM formulation, which is originally proposed by Bradley and Mangasarian [4] not for the ranking but for the classification problem, where the 1-norm of the weight vector is regularized. This version of Ranking SVM is called the 1-norm Ranking SVM. It has some advantages over the standard Ranking SVM: It is formulated as a linear programming (LP) problem and thus can be solved much faster if its size is not too large. Moreover, note that again, the resulting weight vector tends to be sparse and thus is suitable for the feature selection. Unfortunately, the LP problem for the 1-norm Ranking SVM is naturally of size $O(pn(pn+N))$, the same as the QP problem for the standard Ranking SVM. However, in this case, it is unclear how the techniques used in SVM-Perf and PRSVM are applied for solving the LP problem efficiently.

To avoid the difficulty, we take a different approach by simplifying the LP formulation for the 1-norm Ranking SVM, rather than devising a fast algorithm for it. More precisely, we reformulate the LP problems for hard as well as soft margin optimization, with additional constraints that the dual variables d_{ij} are restricted to the product form $d_{ij} = d_i^+ d_j^-$, where i and j range over the p positive and n negative instances, respectively. With these constraints, the number of variables is reduced from pn to $p+n$, which results in greatly simplified optimization problems of size $O((p+n)(p+n+N))$. We call the simplified problems $\mathrm{OPT}_{\mathrm{hard}}$ and $\mathrm{OPT}_{\mathrm{soft}}$ that correspond to the hard and soft margin optimization, respectively. Apparently, they would have worse solutions (i.e., weight vectors with less margins) than the original hard and soft margin 1-norm Ranking SVMs, because the additional constraints significantly reduce the feasible solution spaces. However, surprisingly, $\mathrm{OPT}_{\mathrm{hard}}$ turns out to be equivalent to the original hard margin 1-norm Ranking SVM. In other words, the optimal solution (d_{ij}^*) of the hard margin 1-norm Ranking SVM and the optimal solution $(\hat{d}_i^+, \hat{d}_j^-)$ of $\mathrm{OPT}_{\mathrm{hard}}$ actually have the relation $d_{ij}^* = \hat{d}_i^+ \hat{d}_j^-$. This motivates the reformulation for the soft margin optimization, i.e., $\mathrm{OPT}_{\mathrm{soft}}$.

Unfortunately, unlike in the case of the hard margin optimization, the equivalence between $\mathrm{OPT}_{\mathrm{soft}}$ and the soft margin 1-norm Ranking SVM does not hold. To make matters worse, $\mathrm{OPT}_{\mathrm{soft}}$ is not a convex problem, let alone an LP problem. Nevertheless, we show that a feasible solution can be found by solving an LP problem that is obtained from $\mathrm{OPT}_{\mathrm{soft}}$ by fixing a parameter, and the solution has a certain amount of margin. Furthermore, if the given sample is close to be linearly separable, then our theoretical guarantee on the margin becomes close to that of the soft margin 1-norm Ranking SVM. We also give a practical heuristic to improve the feasible solution up to a local optimum.

Although, as mentioned above, several efficient algorithms for the 2-norm Ranking SVM have been proposed [7], [17], we show that our reformulation technique can be extended to the formulation of the 2-norm Ranking SVM, which yield simplified QP problems of size $O((p+n)N)$ in the primal form and of size $O(pn)$ in the dual form.

We conduct several experiments using artificial and real data sets. Surprisingly, the results show that our methods not only run much faster than most of the previously proposed algorithms as expected, they achieve relatively high AUC scores for many data sets.

Several related works have been done in the literature. Yu and Kim proposed a different notion of 1-norm Ranking SVM [33], where the weight vector is restricted to a linear combination of support vectors, so that the dual form of its LP formulation can be kernelized. However, then its size is $O((pn)^2)$, which is an unacceptable blowup in size. Moreover, in their formulation the 1-norm of the dual variables (the coefficients of linear combination) is regularized, and so the resulting weight vector is not always sparse. Another approach to maximizing the margin with a sparse weight vector is to use the boosting technique such as the AdaBoost [13], [24], [25], although in the naive implementation we would again face with the same obstacle that the pair-sample constructed through the reduction is of size pn. Freund et al. proposed the RankBoost [12] that simulates the AdaBoost over the pair-sample in time linear in the original sample size $(p+n)$. Later, Rudin and Schapire showed that under certain assumptions, the AdaBoost is equivalent to the RankBoost [27]. However, the RankBoost has only a (weak) guarantee of hard margin and no theoretical justification is given when the sample is not linearly separable. Moribe et al. proposed the SoftRankBoost [22] based on the smooth boosting framework [3], [10], [14], [15], [29], so that it works well when the sample is not linearly separable. The SoftRankBoost runs in $O((p+n)N)$ time per iteration and is shown to have the same guarantee of soft margin as that of our algorithm. But the SoftRankBoost does not seem to solve a soft margin optimization problem in any sense.

We would like to mention that the notion of AUC consistency is recently proposed for a criterion of ranking algorithms. An algorithm is said to be AUC consistent if the algorithm outputs a ranking function that converges the Bayes optimal one, that is, the ranking function that maximizes the AUC with respect to the underlying probability distribution. Uematsu et al. [18] first propose this notion and investigate the relation between AUC maximization and a convex loss minimization, where the loss function is an up-

per bound on the ranking risk. In particular, they show that the algorithm for minimizing hinge loss is not AUC consistent. Kotolowski et al. [19] and Agarwal [1] show that the algorithms for minimizing exponential loss and logistic loss are both AUC consistent. These results would suggest that as for AUC maximization the AdaBoost (minimizing exponential loss) works well but the Ranking SVM (minimizing hinge loss) does not. However, in these results, the ranking functions are assumed to be chosen from the universal hypothesis class containing all functions (not only linear functions), and they do not discuss the performance of algorithms when the hypothesis class is restricted to, say, linear functions.

The rest of this paper is organized as follows: In Sect. 2, the problem setting and a known theorem of the generalization bound for AUC are described. In Sect. 3, we show the formulation of 1-norm Ranking SVM. In Sect. 4, we show the equivalence between 1-norm hard margin Ranking SVM and 1-norm hard margin classification SVM with bias. In Sect. 5, we provide our reformulation of soft margin 1-norm Ranking SVM. In Sect. 6, we provide an algorithm for the reformulated optimization problem approximately. In Sect. 7, we experiment evaluate the performance of our method on AUC and computation time comparing various methods. In Sect. 8, we mention the conclusion of this paper.

2. Preliminaries

Let $X \subseteq \mathbb{R}^N$ be an instance space. X is partitioned into two classes: X^+ the class of positive, X^- the class of negative. A ranking function is a function that maps X to $\mathbb{R}$ and let H be a class of ranking functions. The learner is given a sample S that consists of p positive instances $\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_p^+ \in X$ and n negative instances $\boldsymbol{x}_1^-, \ldots, \boldsymbol{x}_n^- \in X$. Let $m(= p + n)$ denote the number of instances. We assume that instances are independently and identically distributed (i.i.d.) according to some unknown distribution D over X. The goal of the learner is to find a function $h \in H$ that maximizes the *AUC score* given by

$$\mathrm{AUC}(h) = \Pr_{\boldsymbol{x},\boldsymbol{x}' \sim D}\{h(\boldsymbol{x}) > h(\boldsymbol{x}') \mid \boldsymbol{x} \in X^+, \boldsymbol{x}' \in X^-\}.$$

For $\rho > 0$, we define the *empirical AUC score of h at ρ* as

$$\mathrm{AUC}_{S,\rho}(h) = \frac{1}{pn} \sum_{i=1}^{p} \sum_{j=1}^{n} I\left(\frac{h(\boldsymbol{x}_i^+) - h(\boldsymbol{x}_j^-)}{2} \geq \rho \right)$$

where $I(\cdot)$ denotes the indicator function.

Throughout the paper, we assume that H is a class of linear functions of the form of $h(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$ for some weight vector $\boldsymbol{w} \in \mathbb{R}^N$. For simplicity, we assume without loss of generality that the weight vectors are non-negative and its 1-norm are normalized. That is,

$$H = \{\boldsymbol{x} \mapsto \boldsymbol{w} \cdot \boldsymbol{x} \mid \boldsymbol{w} \in \mathcal{P}_N\},$$

where $\mathcal{P}_N$ denotes the N-dimensional probability simplex, i.e., $\mathcal{P}_N = \{\boldsymbol{p} \in [0,1]^N \mid \sum_i p_i = 1\}$. To see why we may assume weight vectors to be non-negative, observe that any vector $\boldsymbol{w}$ can be written as $\boldsymbol{w} = \boldsymbol{w}^+ - \boldsymbol{w}^-$ for some two non-negative vectors $\boldsymbol{w}^+$ and $\boldsymbol{w}^-$ such that $\|\boldsymbol{w}\|_1 = \|\boldsymbol{w}^+\|_1 + \|\boldsymbol{w}^-\|_1$. This implies that the reduction $\boldsymbol{x} \mapsto (\boldsymbol{x}, -\boldsymbol{x})$ enables us to learn the general class $\{\boldsymbol{x} \mapsto \boldsymbol{w} \cdot \boldsymbol{x} \mid \|\boldsymbol{w}\| = 1\}$ by a learning algorithm for H (with dimension $2N$).

For the class H of linear functions, Mohri and Rostamizadeh give the following bound on the AUC score in terms of the empirical AUC score [21].

Theorem 1 ([21]). *For any $\rho > 0$ and $\delta > 0$, with probability at least $1 - \delta$, the following bound holds for any linear ranking function $h(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$:*

$$\mathrm{AUC}(h) \geq \mathrm{AUC}_{S,\rho}(h) - \sigma,$$

where

$$\sigma = \sqrt{\frac{4}{mE^4}\left(\frac{8 \ln N}{\rho^2} \ln\left(\frac{4mE^2\rho^2}{\ln N}\right) + 2\ln\left(\frac{2}{\delta}\right)\right)}$$

and

$$E = \mathop{\mathrm{E}}_{x,x' \sim D}[I(x \in X^+, x' \in X^-)].$$

This theorem says that a reasonable way of finding a linear function $h(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$ with high AUC score is to enlarge $\mathrm{AUC}_{S,\rho}(h)$ for as large ρ as possible. Note that the empirical AUC score is rewritten as

$$\mathrm{AUC}_{S,\rho}(h) = \sum_{\boldsymbol{z} \in S'} I(\boldsymbol{w} \cdot \boldsymbol{z} \geq \rho)/|S'|,$$

where $S' = \{(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \mid 1 \leq i \leq p, 1 \leq j \leq n\}$ is called the *pair-sample*. Thus, the problem of finding h with large empirical AUC score can be seen as the standard binary classification problem of finding a large margin classifier $\boldsymbol{w}$ over the pair-sample. For convenience, we will use $\mathrm{AUC}(\boldsymbol{w})$ and $\mathrm{AUC}_{S,\rho}(\boldsymbol{w})$ to denote $\mathrm{AUC}(h)$ and $\mathrm{AUC}_{S,\rho}(h)$ with $h(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$, respectively.

3. 1-Norm Ranking SVM

We use an SVM formulation for the problem of finding a large margin classifier $\boldsymbol{w}$ over the pair-sample. In particular, we use the hard and soft margin SVM formulations with the 1-norm of the weight vector regularized. We call the resultant hard and soft margin optimization problems over the pair-sample the *hard and soft margin 1-norm Ranking SVMs*, respectively. In what follows, we use the following conventions: For an integer u, $[1, u]$ denotes the set $\{1, \ldots, u\}$, and $M = [1, p] \times [1, n]$.

Below we give the primal form of the hard margin 1-norm Ranking SVM.

OP 1: Hard Margin 1-norm Ranking SVM (primal)

$$\max_{\rho, \boldsymbol{w}} \rho$$

sub.to
$$\boldsymbol{w} \cdot (\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \geq \rho, \quad (i,j) \in M,$$
$$\boldsymbol{w} \in \mathcal{P}_N.$$

This is the optimization problem for finding $\boldsymbol{w}$ that maximizes the margin ρ such that $\mathrm{AUC}_{S,\rho}(\boldsymbol{w}) = 1$. So, if the sample is not linearly separable, then we have $\rho < 0$ and Theorem 1 says nothing about $\mathrm{AUC}(\boldsymbol{w})$. The dual form is given as follows.

OP 2: Hard Margin 1-norm Ranking SVM (dual)
$$\min_{\gamma,\boldsymbol{d}} \gamma$$
sub.to
$$\sum_{i,j} d_{ij}(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \leq \gamma \mathbf{1},$$
$$\boldsymbol{d} \in \mathcal{P}_{pn}.$$

Next we give the primal form of the soft margin 1-norm Ranking SVM.

OP 3: Soft Margin 1-norm Ranking SVM (primal)
$$(\rho^*, \boldsymbol{w}^*, \boldsymbol{\xi}^*) = \arg\max_{\rho,\boldsymbol{w},\boldsymbol{\xi}} \rho - \frac{1}{\nu pn}\sum_{i=1}^{p}\sum_{j=1}^{n}\xi_{ij}$$
sub.to
$$\boldsymbol{w} \cdot (\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \geq \rho - \xi_{ij}, \quad (i,j) \in M$$
$$\boldsymbol{w} \in \mathcal{P}_N,$$
$$\xi_{ij} \geq 0, \quad (i,j) \in M,$$

where $\nu \in (0,1]$ is a parameter. Intuitively, this optimization problem is for finding $\boldsymbol{w}$ that maximizes the target margin ρ as well as minimizes the sum of the hinge losses ξ_{ij}, i.e., the quantity by which the instance $(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2$ in the pair-sample violates the target margin ρ. Here the parameter ν controls the tradeoff between the two objectives. By using the KKT conditions, we can show that the optimal solution guarantees that the number of indices $(i,j) \in M$ for which $\boldsymbol{w}^* \cdot (\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \leq \rho^*$ is at most νpn [28], [32]. In other words, we have
$$\mathrm{AUC}_{S,\rho^*}(\boldsymbol{w}^*) \geq 1 - \nu. \tag{1}$$

Note that ρ^* depends on ν and becomes positive when ν is small, even if the sample is not linearly separable. So, the soft margin 1-norm Ranking SVM is quite a robust approach for obtaining a linear function with high AUC score.

The dual problem is given as

OP 4: Soft Margin 1-norm Ranking SVM (dual)
$$(\gamma^*, \boldsymbol{d}^*) = \arg\min_{\gamma,\boldsymbol{d}} \gamma$$
sub.to
$$\sum_{i,j} d_{ij}(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \leq \gamma \mathbf{1},$$
$$d_{ij} \leq \frac{1}{\nu pn}, \quad (i,j) \in M$$
$$\boldsymbol{d} \in \mathcal{P}_{pn}.$$

Note that the size of the soft margin 1-norm Ranking SVM is $O(pn(pn+N))$.

4. Reformulation of the Hard Margin 1-Norm Ranking SVM

In this section, we reformulate the hard margin 1-norm Ranking SVM by restricting the dual variables $\boldsymbol{d} \in \mathcal{P}_{pn}$ to the product form $d_{ij} = d_i^+ d_j^-$, where $\boldsymbol{d}^+ \in \mathcal{P}_p$ and $\boldsymbol{d}^- \in \mathcal{P}_n$ are new variables. Then, since
$$\begin{aligned}
&\sum_{i,j} d_{ij}(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \\
&= \sum_{i,j} d_i^+ d_j^-(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \\
&= \sum_i d_i^+\Big(\sum_j d_j^-\Big)\boldsymbol{x}_i^+/2 - \sum_j d_j^-\Big(\sum_i d_i^+\Big)\boldsymbol{x}_j^-/2 \\
&= \sum_i d_i^+ \boldsymbol{x}_i^+/2 - \sum_j d_j^- \boldsymbol{x}_j^-/2,
\end{aligned}$$
we obtain from OP 2 the following simplified LP problem, called $\mathrm{OPT}_{\mathrm{hard}}$.

OP 5: $\mathrm{OPT}_{\mathrm{hard}}$ (dual)
$$\min_{\gamma,\boldsymbol{d}^+,\boldsymbol{d}^-} \gamma$$
sub.to
$$\sum_i d_i^+ \boldsymbol{x}_i^+/2 - \sum_j d_j^- \boldsymbol{x}_j^-/2 \leq \gamma \mathbf{1},$$
$$\boldsymbol{d}^+ \in \mathcal{P}_p,\ \boldsymbol{d}^- \in \mathcal{P}_n.$$

It turns out that the primal form of the LP problem above is the standard classification version of hard margin 1-norm SVM over the original sample S, which now has the bias term:

OP 6: $\mathrm{OPT}_{\mathrm{hard}}$ (primal)
$$\max_{\rho,\boldsymbol{w},b} \rho$$
sub.to
$$\boldsymbol{w} \cdot \boldsymbol{x}_i^+ + b \geq \rho, \quad i \in [1,p]$$
$$\boldsymbol{w} \cdot \boldsymbol{x}_j^- + b \leq -\rho, \quad j \in [1,n]$$
$$\boldsymbol{w} \in \mathcal{P}_N.$$

Note that $\mathrm{OPT}_{\mathrm{hard}}$ is of size $O((p+n)N)$ in the both forms.

In the following, we show that $\mathrm{OPT}_{\mathrm{hard}}$ is equivalent to the original hard margin 1-norm Ranking SVM (OP 1), by showing that an optimal solution of OP 1 can be constructed from an optimal solution of OP 6[†].

[†]Simlilar results are seen in other literature (e.g., [31], [34]). However, in our knowledge, the previous version of this paper is the original [30], and this theorem is not the main claim of this paper.

Theorem 2. *Let* $(\rho_b, \boldsymbol{w}_b, b_b)$ *be an optimal solution of* OP 6. *Then,* $(\rho_b, \boldsymbol{w}_b)$ *is also an optimal solution of* OP 1.

Proof. Let $(\rho_p, \boldsymbol{w}_p)$ be an optimal solution of OP 1. Clearly, $(\rho_b, \boldsymbol{w}_b)$ is a feasible solution of OP 1. Hence, $\rho_b \leq \rho_p$. Next, we show that the opposite is true. Let $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ be positive and negative *support vectors* of OP 1 for which $\boldsymbol{w}_p \cdot (\boldsymbol{x}^+ - \boldsymbol{x}^-)/2 = \rho_p$. Let

$$b_p = -\boldsymbol{w}_p \cdot (\boldsymbol{x}^+ + \boldsymbol{x}^-)/2.$$

Then, $(\rho_p, \boldsymbol{w}_p, b_p)$ is a feasible solution of OP 6. To see this, for any positive instance $\boldsymbol{x}_i^+$, observe that

$$\begin{aligned}\boldsymbol{w}_p \cdot \boldsymbol{x}_i^+ + b_p &= \boldsymbol{w}_p \cdot (\boldsymbol{x}_i^+ - \boldsymbol{x}^-)/2 + \boldsymbol{w}_p \cdot (\boldsymbol{x}_i^+ - \boldsymbol{x}^+)/2 \\ &\geq \rho_p + \frac{\boldsymbol{w}_p \cdot (\boldsymbol{x}_i^+ - \boldsymbol{x}^-) - \boldsymbol{w}_p \cdot (\boldsymbol{x}^+ - \boldsymbol{x}^-)}{2} \\ &\geq \rho_p + \rho_p - \rho_p = \rho_p.\end{aligned}$$

A similar inequality holds for negative instances as well. So we have $\rho_p \leq \rho_b$. □

5. Reformulation of the Soft Margin 1-Norm Ranking SVM

Motivated by the equivalence result of the hard margin case, we now reformulate the soft margin 1-norm Ranking SVM with the same additional constraints $d_{ij} = d_i^+ d_j^-$. Then we obtain from OP 4 the following simplified optimization problem:

OP 7: $\mathrm{OPT}_{\mathrm{soft}}$

$$\begin{aligned}\hat{\gamma} = \min_{\gamma, \boldsymbol{d}^+, \boldsymbol{d}^-, \nu^+} \ & \gamma \\ \text{sub.to} \ & \\ & \sum_i d_i^+ \boldsymbol{x}_i^+/2 - \sum_j d_j^- \boldsymbol{x}_j^-/2 \leq \gamma \mathbf{1}, \\ & d_i^+ \leq \frac{1}{\nu^+ p}, \quad i \in [1, p] \\ & d_j^- \leq \frac{\nu^+}{\nu n}, \quad j \in [1, n] \\ & \boldsymbol{d}^+ \in \mathcal{P}_p, \ \boldsymbol{d}^- \in \mathcal{P}_n.\end{aligned}$$

Note that we replace the constraint $\max_{(i,j)\in M} d_{ij} \leq 1/\nu pn$ of OP 4 by $\max_i d_i^+ \max_j d_j^- \leq 1/\nu pn$, which is further replaced by the two constraints $\max_i d_i^+ \leq 1/\nu^+ p$ and $\max_j d_j^- \leq \nu^+/\nu n$ with a new variable ν^+ to be optimized.

$\mathrm{OPT}_{\mathrm{soft}}$ is of size $O((p+n)(p+n+N))$ and thus seems to be easier to solve. But it is not a convex optimization problem since the constraints $d_i^+ \leq 1/\nu^+ p$ are not convex. To overcome this difficulty, we first consider $\mathrm{OPT}_{\mathrm{soft}}$ as an LP problem with ν^+ to be fixed to a constant. The LP problem with parameter ν^+ is called $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$:

OP 8: $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ (dual)

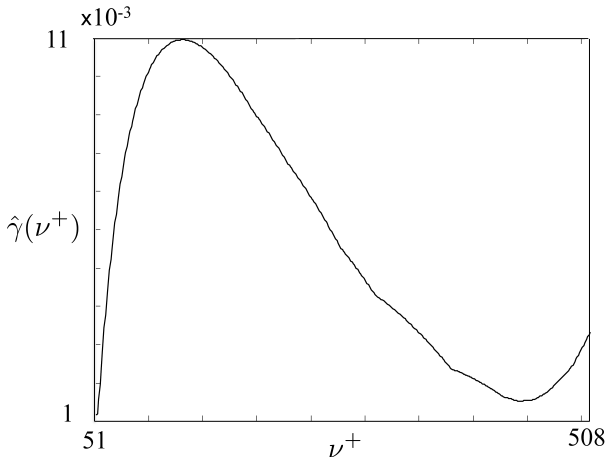

Fig. 2 Non-convexity of the function $\hat{\gamma}(\nu^+)$ for an artificial data set.

$$\begin{aligned}\hat{\gamma}(\nu^+) = \min_{\gamma, \boldsymbol{d}^+, \boldsymbol{d}^-} \ & \gamma \\ & \text{sub.to the same constraints as in } \mathrm{OPT}_{\mathrm{soft}}\end{aligned}$$

Clearly, $\min_{\nu^+} \hat{\gamma}(\nu^+) = \hat{\gamma}$. Unfortunately, the function $\hat{\gamma}(\nu^+)$ is not convex with respect to ν^+ (see Fig. 2 for example). So, it seems to be hard to obtain the optimum. In the next section we propose an iterative linearization-minimization method to find a local optimal solution of ν^+.

On the other hand, for any fixed choice of ν^+, we can guarantee that the solution of $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ has a certain amount of empirical AUC score. To see this, we first give the primal form of $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$:

OP 9: $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ (primal)

$$\begin{aligned}&(\hat{\rho}, \hat{\boldsymbol{w}}, \hat{b}, \hat{\boldsymbol{\xi}}^+, \hat{\boldsymbol{\xi}}^-) \\ &= \arg \max_{\rho, \boldsymbol{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-} \rho - \frac{1}{2\nu^+ p} \sum_i \xi_i^+ - \frac{\nu^+}{2\nu n} \sum_j \xi_j^- \\ &\text{sub.to} \\ &\boldsymbol{w} \cdot \boldsymbol{x}_i^+ + b \geq \rho - \xi_i^+, \quad i \in [1, p] \\ &-\boldsymbol{w} \cdot \boldsymbol{x}_j^- - b \geq \rho - \xi_j^-, \quad j \in [1, n] \\ &\boldsymbol{w} \in \mathcal{P}_N, \ \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \geq \mathbf{0}.\end{aligned}$$

Theorem 3. *For a fixed* ν^+, *let* $\hat{\rho}$ *and* $\hat{\boldsymbol{w}}$ *be the solutions of* $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ (primal). *Then,*

$$\mathrm{AUC}_{S,\hat{\rho}}(\hat{\boldsymbol{w}}) \geq 1 - \nu^+ - \frac{\nu}{\nu^+} + \nu.$$

Proof. By using the KKT conditions, we have $\hat{\xi}_i^+(\hat{d}_i^+ - 1/\nu^+) = 0$. So, if $\hat{\xi}_i^+ > 0$ then $\hat{d}_i^+ = 1/\nu^+ p$. Since there are at most $\nu^+ p$ indices i such that $\hat{d}_i^+ = 1/\nu^+ p$, there are at most $\nu^+ p$ indices i with $\hat{\xi}_i^+ > 0$. Similarly, there are at most $\nu n/\nu^+$ indices j with $\hat{\xi}_j^- > 0$. Therefore, for at least $(p - \nu^+ p)(n - \nu n/\nu^+)$ pairs of instances $\boldsymbol{z}_{ij}$ in the pair-sample, $\hat{\boldsymbol{w}} \cdot \boldsymbol{z} \geq \hat{\rho}$. □

For particular choices of the parameters ν and ν^+, we

obtain the following corollary.

Corollary 4. *Let $\hat{\rho}$ and $\hat{\boldsymbol{w}}$ be the optimal solution of* $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ (primal) *for* $\nu^+ = \sqrt{\nu}$. *Then,*

$$\mathrm{AUC}_{S,\hat{\rho}}(\hat{\boldsymbol{w}}) \geq (1 - \sqrt{\nu})^2.$$

For comparison, we show the guarantee (1) of the original soft margin 1-norm Ranking SVM for the same choice of ν as in the corollary above:

$$\mathrm{AUC}_{S,\rho^*}(\boldsymbol{w}^*) \geq 1 - \nu.$$

Below we compare ρ^* and $\hat{\rho}$. Note that, by duality we have

$$\hat{\rho} - \frac{1}{2\nu^+ p}\sum_i \hat{\xi}_i^+ - \frac{\nu^+}{2\nu n}\sum_j \hat{\xi}_j^- = \hat{\gamma}(\nu^+).$$

Combined this with the fact that

$$\hat{\gamma}(\nu^+) \geq \hat{\gamma} \geq \gamma^* = \rho^* - (1/\nu pn)\sum_{ij} \xi_{ij}^*,$$

we have

$$\hat{\rho} \geq \rho^* - \frac{1}{\nu pn}\sum_{ij} \xi_{ij}^*.$$

Therefore, if $\sum_{ij} \xi_{ij}^*$ is small, i.e., the sample is close to be linearly separable, then we can say that the solution $\hat{\boldsymbol{w}}$ of $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ has nearly as high AUC score as the original 1-norm soft margin Ranking SVM.

6. An Iterative Linearization-Minimization Method for Optimizing ν^+

Now we give an iterative linearization-minimization method for finding a local optimal solution of ν^+, which attains a local minimum of $\mathrm{OPT}_{\mathrm{soft}}$. Recall that in $\mathrm{OPT}_{\mathrm{soft}}$, we have non-convex constraints $d_i^+ \leq 1/\nu^+ p$. In order to make them convex, we replace them by their linear approximations. To be more precise, we consider the constraint that every d_i^+ is bounded by the tangent line of $1/\nu^+ p$ at some point $\nu^+ = \nu_c^+$. That is,

$$d_i^+ \leq -\frac{1}{(\nu_c^+)^2 p}\nu^+ + \frac{2}{\nu_c^+ p}. \tag{2}$$

Thus we have the following LP problem called $\mathrm{LP}_{\mathrm{soft}}$, where ν_c^+ is a parameter:

OP 10: $\mathrm{LP}_{\mathrm{soft}}$

$$(\tilde{\gamma}, \tilde{\boldsymbol{d}}^+, \tilde{\boldsymbol{d}}^-, \tilde{\nu}^+) = \arg\min_{\boldsymbol{d}^+, \boldsymbol{d}^-, \gamma, \nu^+} \gamma$$

sub.to

$$\sum_i d_i^+ \boldsymbol{x}_i^+/2 - \sum_j d_j^- \boldsymbol{x}_j^-/2 \leq \gamma \mathbf{1},$$

$$d_i^+ \leq -\frac{1}{(\nu_c^+)^2 p}\nu^+ + \frac{2}{\nu_c^+ p}, \quad i \in [1, p]$$

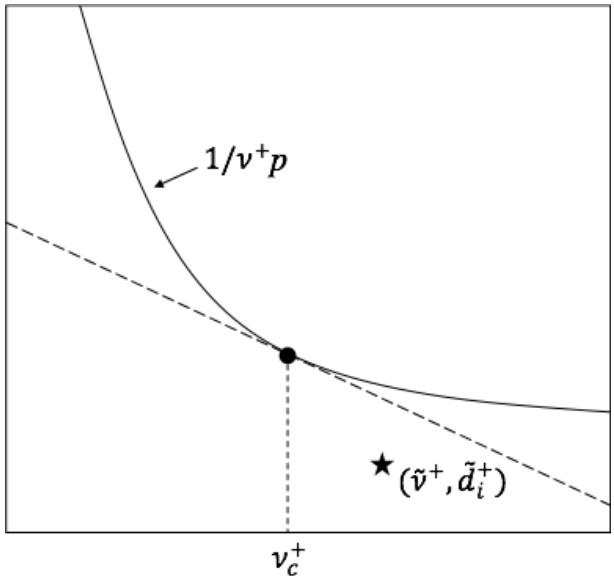

Fig. 3 Illustration of our algorithm. The dotted line shows the tangent line of $1/\nu^+ p$ at $\nu^+ = \nu_c^+$. $\tilde{\nu}^+$ is the solution of $\mathrm{LP}_{\mathrm{soft}}$.

$$d_j^- \leq \frac{\nu^+}{\nu n}, \quad j \in [1, n]$$

$$\boldsymbol{d}^+ \in \mathcal{P}_p,\ \boldsymbol{d}^- \in \mathcal{P}_n.$$

Note that since any d_i^+ satisfying the new constraint (2) also satisfies the original constraints $d_i^+ \leq 1/\nu^+ p$, the optimal solution of $\mathrm{LP}_{\mathrm{soft}}$ is a feasible solution of $\mathrm{OPT}_{\mathrm{soft}}$.

Now we are ready to describe our algorithm:

1. Let ν_c^+ be an initial guess.
2. Solve $\mathrm{LP}_{\mathrm{soft}}$ and get a solution $(\tilde{\boldsymbol{d}}^+, \tilde{\boldsymbol{d}}^-, \tilde{\gamma}, \tilde{\nu}^+)$.
3. If the value of $\tilde{\gamma}$ decreases, then let $\nu_c^+ = \tilde{\nu}^+$ and go to 2.
4. Solve $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$(dual) with $\nu^+ = \tilde{\nu}^+$ and get a solution $(\hat{\boldsymbol{d}}^+, \hat{\boldsymbol{d}}^-, \hat{\gamma})$.

A reasonable choice of the initial guess in the first step would be $\nu_c^+ = \sqrt{\nu}$, as in Corollary 4. Observe that the solution $(\tilde{\boldsymbol{d}}^+, \tilde{\boldsymbol{d}}^-, \tilde{\gamma}, \tilde{\nu}^+)$ of $\mathrm{LP}_{\mathrm{soft}}$ is a feasible solution of $\mathrm{LP}_{\mathrm{soft}}(\tilde{\nu}^+)$. So, the minimum $\tilde{\gamma}'$ of $\mathrm{LP}_{\mathrm{soft}}(\tilde{\nu}^+)$ satisfies $\tilde{\gamma}' \leq \tilde{\gamma}$. Therefore, by repeating this procedure, we can obtain a monotonically decreasing sequence of $\tilde{\gamma}$, which will converge to a local minimum. Figure 3 illustrates the algorithm. Note that the final step is redundant and thus can be skipped. We add this just for numerical stability.

7. Experiments

In the following experiments, we verify effectiveness and efficiency of our method for maximizing AUCs. The data sets include artificial data sets, and some real data sets.

7.1 Artificial Data

For the first experiment, we used artificial data sets with r-of-k threshold functions as target functions. An r-of-k

Table 1 AUCs for artificial data sets with random noises 5%, 10%, and 15% so that ratios of positive and negative instances are 5:5.

data		SVM-Perf	1-norm C. SVM	Rank Boost	Soft RankBoost	Original 1-norm	**our method**	
r	noise						**1-norm**	2-norm
1	5(%)	0.9363	0.9259	0.9480	0.6991	0.9380	0.9445	**0.9566**
8		0.8967	**0.9607**	0.9285	0.9441	0.9592	0.9601	0.9566
15		0.9325	**0.9547**	0.9236	0.9537	0.9515	0.9526	0.9374
1	10(%)	0.8951	0.3469	0.9030	0.6459	0.8909	0.9071	**0.9209**
8		0.8658	0.9166	0.8928	**0.9227**	0.9186	0.9221	0.9169
15		0.8862	0.8929	0.8786	**0.9158**	0.9002	0.9044	0.8914
1	15(%)	0.8480	0.0455	0.8338	0.6343	0.6396	0.8343	**0.8650**
8		0.8337	0.8595	0.8516	0.8566	0.8663	**0.8730**	0.8687
15		0.8436	0.8307	0.8359	0.8598	0.8450	**0.8613**	0.8347

Table 2 AUCs for artificial data sets so that ratios of positive and negative instances are 7:3, 9:1.

data		SVM-Perf	1-norm C. SVM	Rank Boost	Soft RankBoost	Original 1-norm	**our method**	
$p:n$	r						**1-norm**	2-norm
7:3	1	0.9411	0.9339	0.9257	0.8440	0.9231	0.9046	**0.9458**
	8	0.9166	**0.9397**	0.9078	0.9317	0.9333	0.9336	0.9177
	15	0.9179	0.9344	0.9027	**0.9392**	0.9282	0.9333	0.9093
9:1	1	0.7950	0.3724	0.8014	0.8095	0.8049	**0.8175**	0.7974
	8	0.7659	**0.8180**	0.7734	0.7773	0.7678	0.7748	0.7645
	15	0.7269	**0.7946**	0.7567	0.7573	0.7494	0.7774	0.7578

threshold function f over N Boolean variables is associated with some set A of k Boolean variables and f outputs $+1$ if at least r of the k variables in A are positive and f outputs -1, otherwise. Assume that the instance space is $\{+1,-1\}^N$. That is, the r-of-k threshold function f is represented as

$$f(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{x\in A} x + k - 2r + 1\right).$$

For $N = 100$, $k = 30$, and $r = 1, 8, 15$, we fix r-of-k threshold functions which determine labels. Then for each set of parameters, we generate $m = 1000$ random instances so that ratios of positive and negative instances are $5:5$, $7:3$, and $9:1$ respectively. Finally, we add random noise into labels by changing the label of each instance with probability 5%, 10%, and 15%. As hypotheses, we use N Boolean variables themselves and the constant hypothesis which always outputs $+1$.

We compare RankBoost [12], SoftRankBoost [22], original 1-norm Ranking SVM (which solves OP 3 and OP 4, and we call "*Original 1-norm*"), and our methods. We also compare with classification version of soft margin 1-norm SVM [4] (we call 1-norm C. SVM for short), which solves soft margin version of OP 6. Additionaly, we also compare with 2-norm version of our method[†] and SVM-Perf [17] which efficiently simulates 2-norm (standard) Ranking SVM. Note that the experimental results of 2-norm methods are expletive since our method of 2-norm Ranking SVM work efficiently more than original Ranking SVM [16] but not efficiently more than SVM-Perf or the other latest methods (e.g. [7]). The main goal of this paper is to solve 1-norm Ranking SVM efficiently.

[†]The 2-norm version of our method and the formulations are described in Appendix.

For RankBoost, we set the number of iterations as $T = 1000, 10000, 100000$. For the other methods, we set the parameter $\nu \in \{0.05, 0.1, 0.15, 0.2\}$. For SVM-Perf, we set the parameter $\epsilon = 0.001$, $C = 10, 20, \ldots, 100$. We evaluate each method by 5-fold cross validation. Table 1 is the result that we change the noises, keep $p:n = 5:5$. Table 2 is the result that we change the ratios $p:n$, keep noises 5%. Surprisingly, our methods achieve high AUC scores comparing with the other methods. Particularly, contrast to our theoretical speculation (see Corollary 4), our method of 1-norm often beats Original 1-norm in practice. It is not easy to explain the reason, however, we think that our methods may avoid overfitting through simplifying. We can observe that SoftRankBoost also often achieves high AUC scores, our methods perform better in stably. 1-norm C. SVM sometimes achieves high AUC scores, however, the performance is highly unstable. It is considered to be due to weakness against noise. Therefore, also in practice, we cannot say that classification SVM has robustness for bipartite ranking problem.

7.2 Real Data

For the next experiment we use data sets "hypothyroid", "ionosphere", "australian", "colon cancer" and "duke breast cancer" in LIBSVM data [6]. We set the parameter C of SVM-Perf $100, 200, \ldots, 1000$. The parameters of the other algorithms are the same as in Sect. 7.1. As can be seen in Table 3, It is not to say that our methods are clearly better than the other methods, but our methods stably archive high AUCs for all data sets.

7.3 Computation Time

In this experiment, we will compare our method of 1-norm

Table 3 AUCs for LIBSVM data sets.

data	SVM-Perf	1-norm C. SVM	Rank Boost	Soft RankBoost	Original 1-norm	**our method** **1-norm**	2-norm
hypothyroid	**0.9374**	0.6661	0.7883	0.8504	0.8504	0.9091	0.9318
ionosphere	0.9643	0.9282	0.8961	0.9518	0.9778	**0.9790**	0.9568
australian	0.8882	0.8980	0.9171	0.8896	0.9298	0.9257	**0.9325**
colon-cancer	0.8200	0.7850	0.8500	**0.9350**	0.7900	0.8950	0.9000
duke	0.9520	0.9520	0.6720	0.8720	**0.9600**	0.9520	0.9120

Table 4 Computation time (sec.).

m	SVM-Perf $C = 100$	SVM-Perf $C = 10000$	1-norm C. SVM	Rank Boost	Soft RankBoost	Original 1-norm	**our method** **1-norm**	2-norm
250	4.2930	17.20	0.0884	0.4039	0.1283	7.9	0.4204	2.5001
500	3.4043	245.21	0.1879	0.0682	0.1313	35.3	1.5280	7.7216
1000	4.6884	524.81	0.6226	0.0514	0.1934	149.3	2.8103	34.0069
1500	6.5211	356.32	1.0289	0.0734	0.2971	391.2	7.9647	70.8608
3000	9.5591	771.55	3.3639	0.0978	0.3606	1320.0	8.6830	288.0392

Table 5 The number of non-zero features of weight vectors obtained by each method using LIBSVM data sets. N is the dimension each data set has.

data	N	SVM-Perf	1-norm C. SVM	Rank Boost	Soft RankBoost	Original 1-norm	**our method** **1-norm**	2-norm
hypothyroid	43	29	16	12	2	32	2	28
ionosphere	34	33	9	12	11	7	9	32
australian	14	8	7	6	6	9	8	14
colon-cancer	2000	1997	29	159	4	27	18	1991
duke	7129	7048	25	375	21	25	25	7066

to 1-norm C. SVM Original 1-norm and boosting methods, and compare our method of 2-norm to SVM-Perf. The time complexity of SVM-Perf is guaranteed $O(sm\log(m))$, where s is the number of non-zero features. We use the machine with 16 cores of Intel Xeon 5560 2.80GHz and 198GByte memory, and use the artificial data sets, the size of each data set is $m = 250, 500, 1000, 1500, 3000$, respectively. The parameters of SVM-Perf are $\epsilon = 0.001$, $C = 100$ and 10000. For RankBoost, we set $T = 100$. We evaluate each execution time which is consumed to train for 5-fold cross validation and is averaged.

In 1-norm case, as is shown in Table 4, our method is clearly faster than Original 1-norm. However, RankBoost and SoftRankBoost is much faster than our method. 1-norm C. SVM is faster than our method because our method iteratively runs as described in Sect. 6. However, the difference does not depend on sample size. Also in 2-norm case, our method is faster than SVM-Perf set by $C = 10000$. Since the computation time of SVMPerf depends on parameter C (see [17] lemma 2), we have to take time to find the best parameter C for any data sets. Our method stably runs regardless of each parameter ϵ.

7.4 Sparsity

Finally, we show that the hyperplane obtained by 1-norm regularized methods has high sparsity using LIBSVM data sets. As seen in Table 5, 1-norm C. SVM, SoftRankBoost, Original 1-norm and our method of 1-norm obtain sparse weight vectors for large feature data sets, and weight vectors of our method of 1-norm are as sparse as those of Original 1-norm. Note that the each norm of weight vector of SVM-Perf, RankBoost, and our method of 2-norm are normalized to 1.

8. Conclusion and Future Work

In this paper, we have reformulated the Ranking SVMs for ranking functions as significantly simplified optimization problems of size $O(m^2)$, where m is the size of the original sample. We give theoretical guarantees on the generalization ability of the ranking functions obtained by solving the optimization problems. In particular, the reformulation of the 1-norm Ranking SVM yields the first practical algorithm that is competitive with the original 1-norm Ranking SVM in performance.

As future work, we apply our practical method to optimizing other criteria biased to top elements [23], [26].

References

[1] S. Agarwal, "Surrogate regret bounds for bipartite ranking via strongly proper losses," CoRR, abs/1207.0268, 2012.
[2] M.-F. Balcan, N. Bansal, A. Beygelzimer, D. Coppersmith, J. Langford, and G.B. Sorkin, "Robust reductions from ranking to classification," Proceedings of the 20th Annual Conference on Learning Theory, pp.604–619, 2007.
[3] J.K. Bradley and R. Shapire, "FilterBoost: Regression and classification on large datasets," Advances in Neural Information Processing Systems 20, pp.185–192, 2008.
[4] P. Bradley and O.L. Mangasarian, "Feature selection via concave

minimization and support vector machines," Machine Learning Proceedings of the 15th International Conference, pp.82–90, Morgan Kaufmann, 1998.
[5] U. Brefeld and T. Scheffer, "AUC maximizing support vector learning," Proceedings of the ICML Workshop on ROC Analysis in Machine Learning, 2005.
[6] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol.2, no.3, pp.27:1–27:27, 2011.
[7] O. Chapelle and S.S. Keerthi, "Efficient algorithms for ranking with SVMs," Inf. Retr., vol.13, no.3, pp.201–215, June 2010.
[8] W.W. Cohen, R.E. Schapire, and Y. Singer, "Learning to order things," Journal of Artificial Intelligence Research, vol.10, pp.243–279, 1999.
[9] C. Cortes and M. Mohri, "AUC optimization vs. error rate minimization," Advances in Neural Information Processing Systems 16, 2004.
[10] C. Domingo and O. Watanabe, "MadaBoost: A Modification of AdaBoost," Proceedings of 13th Annual Conference on Computational Learning Theory, pp.180–189, 2000.
[11] K. Duh and K. Kirchhoff, "Learning to rank with partially-labeled data," Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, pp.251–258, ACM, 2008.
[12] Y. Freund, R. Iyer, R.E. Shapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," Journal of Machine Learning Research, vol.4, pp.933–969, 2003.
[13] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol.55, no.1, pp.119–139, 1997.
[14] D. Gavinsky, "Optimally-smooth adaptive boosting and application to agnostic learning," Journal of Machine Learning Research, vol.2533, pp.98–112, 2003.
[15] K. Hatano, "Smooth boosting using an information-based criterion," Proceedings of the 17 th International Conference on Algorithmic Learning Theory, vol.4264, pp.304–319, 2006.
[16] T. Joachims, "Optimizing search engines using clickthrough data," Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, 2002.
[17] T. Joachims, "A support vector method for multivariate performance measures," Proceedings of the 22nd international conference on Machine learning, New York, NY, USA, pp.377–384, ACM, 2005.
[18] K. Uematsu and Y. Lee, "On theoretically optimal ranking functions in bipartite ranking," Technical report, Technical Report 863, Department of Statistics, The Ohio State University, vol.112, no.519, pp.1311–1322, 2017.
[19] W. Kotlowski, K.J. Dembczynski, and E. Hüllermeier, "Bipartite ranking through minimization of univariate loss," L. Getoor and T. Scheffer, editors, Proceedings of the 28th International Conference on Machine Learning, New York, NY, USA, pp.1113–1120, ACM, 2011.
[20] P.M. Long and R.A. Servedio, "Boosting the area under the ROC curve," Advances in Neural Information Processing Systems 20, 2008.
[21] A.T. Mehryar Mohri, Afshin Rostamizadeh, "Foundations of Machine Learning," The MIT Press, 2012.
[22] J. Moribe, K. Hatano, E. Takimoto, and M. Takeda, "Smooth boosting for margin-based ranking," Proceedings of the 19th International Conference on Algorithmic Learning Theory, vol.5254, pp.227–239, 2008.
[23] H. Narasimhan and S. Agarwal, "SVM$_{\text{pAUC}}^{\text{tight}}$: A new support vector method for optimizing partial auc based on a tight convex upper bound," Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, New York, NY, USA, pp.167–175, ACM, 2013.
[24] G. Rätsch, "Robust Boosting via Convex Optimization: Theory and Applications," PhD thesis, University of Potsdam, 2001.
[25] G. Rätsch and M.K. Warmuth, "Efficient margin maximizing with boosting," Journal of Machine Learning Research, vol.6, pp.2131–2152, 2005.
[26] C. Rudin, "Ranking with a P-Norm Push," Proceedings of 19th Annual Conference on Learning Theory, vol.4005, pp.589–604, 2006.
[27] C. Rudin and R.E. Schapire, "Margin-based Ranking and an Equivalence between AdaBoost and RankBoost," Journal of Machine Learning Research, vol.10, pp.2193–2232, 2009.
[28] B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett, "New support vector Algorithms," Neural Computation, vol.12, no.5, pp.1207–1245, 2000.
[29] R.A. Servedio, "Smooth boosting and learning with malicious noise," Journal of Machine Learning Research, vol.2111, pp.473–489, 2003.
[30] D. Suehiro, K. Hatano, and E. Takimoto, "Approximate reduction from AUC maximization to 1-norm soft margin optimization," Proceedings of the 22nd International Conference on Algorithmic Learning Theory, vol.6925, pp.324–337, 2011.
[31] A. Takeda, H. Mitsugi, and T. Kanamori, "A unified classification model based on robust optimization," Neural Computation, vol.25, no.3, pp.759–804, March 2013.
[32] M. Warmuth, K. Glocer, and G. Rätsch, "Boosting algorithms for maximizing the soft margin," Advances in Neural Information Processing Systems 20, pp.1585–1592, 2008.
[33] H. Yu, J. Kim, Y. Kim, S. Hwang, and Y.H. Lee, "An efficient method for learning nonlinear Ranking SVM functions," Information Sciences, vol.209, pp.37–48, 2012.
[34] H. Yu and S. Kim, "SVM Tutorial — Classification, Regression and Ranking," Springer Berlin Heidelberg, Berlin, Heidelberg, pp.479–506, 2012.

Appendix: Reformulation of the Soft Margin 2-Norm Ranking SVM

In this section, we employ a similar reformulation and simplification strategy to the standard 2-norm Ranking SVM, although as stated in Introduction, it has efficient algorithms under the original formulation. Here we no longer assume that the weight vector $\boldsymbol{w}$ is in $\mathcal{P}_N$. First we give the standard soft margin 2-norm Ranking SVM, which is based on the ν-SVM formulation [28], where ν-SVM is an equivalent variant of SVM and useful for showing a lower bound on the empirical AUC score[†].

OP 11: Soft Margin 2-norm Ranking SVM (primal)

$$
\begin{aligned}
(\rho^*, \boldsymbol{w}^*, \boldsymbol{\xi}^*) = \min_{\rho, \boldsymbol{w}, \boldsymbol{\xi}} \ & \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \nu\rho + \frac{1}{pn}\sum_{i=1}^{p}\sum_{j=1}^{n}\xi_{ij} \\
& \text{sub.to} \\
& \boldsymbol{w}\cdot(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \geq \rho - \xi_{ij}, \quad (i,j) \in M \\
& \boldsymbol{\xi} \geq \boldsymbol{0},
\end{aligned}
$$

where $0 \leq \nu \leq 1$ is a parameter. By the property of the ν-SVM (see, e.g., [28]), the optimal solution guarantees that the number of pairs $(\boldsymbol{x}_i^+, \boldsymbol{x}_j^-)$ for which $\boldsymbol{w}^*\cdot(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2 \leq \rho^*$ is at most νpn. In other words, we have

[†]In the original formulation [28], there is another constraint that $\rho \geq 0$. In our analysis, we omit the constraint for simplicity.

$$\mathrm{AUC}_{S,\rho^*}(\boldsymbol{w}^*) \geq 1 - \nu. \qquad (\mathrm{A}\cdot 1)$$

The dual form is given below:

OP 12: Soft Margin 2-norm Ranking SVM (dual)

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} -\frac{1}{2}\left\|\sum_{i=1}^{p}\sum_{j=1}^{n}\alpha_{ij}(\boldsymbol{x}_i^+ - \boldsymbol{x}_j^-)/2\right\|^2$$

sub.to

$$0 \leq \alpha_{ij} \leq \frac{1}{pn}, \quad (i,j) \in M$$

$$\sum_{i=1}^{p}\sum_{j=1}^{n}\alpha_{ij} = \nu.$$

Note that the dual form is of size $O((pn)^2)$, assuming that each inner product between instances is of unit size.

Now we give our reformulation. Like the 1-norm case, we replace each dual variable α_{ij} with a (slightly modified) product form $4\alpha_i^+\alpha_j^-/\nu$ but now we put an additional constraint $\sum_i \alpha_i^+ = \sum_j \alpha_j^- (= \nu/2)$. Then, it is easy to see that we obtain from OP 12 the following simplified but non-convex optimization problem, called 2-norm $\mathrm{OPT}_{\mathrm{soft}}$:

OP 13: 2-norm $\mathrm{OPT}_{\mathrm{soft}}$

$$(\widehat{\boldsymbol{\alpha}}^+, \widehat{\boldsymbol{\alpha}}^-, \widehat{\nu}^+)$$

$$= \arg\max_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-,\nu^+} -\frac{1}{2}\left\|\sum_{i=1}^{p}\alpha_i^+\boldsymbol{x}_i^+ - \sum_{j=1}^{n}\alpha_j^-\boldsymbol{x}_j^-\right\|^2$$

sub.to

$$0 \leq \alpha_i^+ \leq \frac{\nu^+}{2p}, \quad i \in [1,p]$$

$$0 \leq \alpha_j^- \leq \frac{\nu}{2n\nu^+}, \quad j \in [1,n]$$

$$\sum_{i=1}^{p}\alpha_i^+ = \sum_{j=1}^{n}\alpha_j^- = \frac{\nu}{2}.$$

Note that we replace the constraint $\max_{(i,j)\in M}\alpha_{ij} \leq 1/(pn)$ of OP 12 by $\max_i \alpha_i^+ \max_j \alpha_j^- \leq \nu/(4pn)$, which is further replaced by the two constraints $\max_i \alpha_i^+ \leq \nu^+/(2p)$ and $\max_j \alpha_j^- \leq \nu/(2n\nu^+)$ with the new variable ν^+ to be optimized.

When we fix ν^+ to a constant, then we have the QP problem, called 2-norm $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$, which has the following primal form:

OP 14: 2-norm $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ (primal)

$$(\hat{\rho}, \hat{\boldsymbol{w}}, \hat{b}, \hat{\boldsymbol{\xi}}^+, \hat{\boldsymbol{\xi}}^-)$$

$$= \arg\min_{\rho,\boldsymbol{w},b,\boldsymbol{\xi}^+,\boldsymbol{\xi}^-} \frac{\|\boldsymbol{w}\|^2}{2} - \nu\rho + \frac{\nu^+}{2p}\sum_{i=1}^{p}\xi_i^+ + \frac{\nu}{2n\nu^+}\sum_{j=1}^{n}\xi_j^-$$

sub.to

$$\boldsymbol{w}\cdot\boldsymbol{x}_i^+ + b \geq \rho - \xi_i, \quad i \in [1,p]$$

$$-\boldsymbol{w}\cdot\boldsymbol{x}_j^- - b \geq \rho - \xi_j, \quad j \in [1,n]$$

$$\boldsymbol{\xi}^+, \boldsymbol{\xi}^- \geq \boldsymbol{0},$$

$$\rho \geq 0.$$

For any fixed choice of ν^+, we can guarantee that the solution of 2-norm $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ has a certain amount of empirical AUC score.

Theorem 5. *For a fixed ν^+, let $\hat{\rho}$ and $\hat{\boldsymbol{w}}$ be the solutions of* 2-norm $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ (primal). Then,

$$\mathrm{AUC}_{S,\hat{\rho}}(\hat{\boldsymbol{w}}) \geq 1 - \nu^+ - \frac{\nu}{\nu^+} + \nu.$$

Proof. By the KKT conditions, $\xi_i^+ > 0$ implies $\alpha_i^+ = \nu^+/(2p)$. Since $\sum_i \alpha_i^+ = \nu/2$, there are at most $\nu p/\nu^+$ indices i such that $\xi_i > 0$. Similarly, there are at most $\nu^+ n$ indiecs j such that $\xi_j > 0$. Therefore, at least $(p - \nu p/\nu^+)(n - \nu^+ n)$ pairs of instances $\boldsymbol{z}_{ij}$ in the pair-sample, $\hat{\boldsymbol{w}}\cdot\boldsymbol{z} \geq \hat{\rho}$. □

For a paricular choice of ν^+, we we obtain the following corollary.

Corollary 6. *Let $\hat{\rho}$ and $\hat{\boldsymbol{w}}$ be the optimal solutions of* 2-norm $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ (primal) for $\nu^+ = \sqrt{\varepsilon}$. Then,

$$\mathrm{AUC}_{S,\hat{\rho}}(\hat{\boldsymbol{w}}) \geq (1 - \sqrt{\nu})^2.$$

Moreover, we can use the iterative linearization-minimization technique to find a local optimal value of ν^+, as we did for the 1-norm case. In this case, we replace the non-convex constraints

$$\alpha_j^- \leq \frac{\nu}{2n\nu^+}$$

of 2-norm $\mathrm{OPT}_{\mathrm{soft}}$ with the linear constraints

$$\alpha_j^- \leq -\frac{\nu}{2n(\nu_c^+)^2}\nu^+ + \frac{\nu}{n\nu_c^+},$$

where ν_c^+ is the current guess. Then, solve the QP problem and obtain $\hat{\nu}^+$. Repeat the procedure above with $\nu_c^+ = \hat{\nu}^+$ until convergence. Finally, solve 2-norm $\mathrm{OPT}_{\mathrm{soft}}(\nu^+)$ with $\nu^+ = \hat{\nu}^+$.

Daiki Suehiro received Ph.D. from Kyushu University in 2014. Currently, he is an assistant professor at Department of Advanced Information Technology, Information Science and Electrical Engineering in Kyushu University and he is also the team member of Computational Learning Theory Team, RIKEN Center for Advanced Intelligence Project. His research interests include machine machine learning theory, data mining, and time-series analysis.

Kohei Hatano received Ph.D. from Tokyo Institute of Technology in 2005. Currently, he is an assosiate professor at Faculty of arts and science in Kyushu University, and he is also the team leader of Computational Learning Theory Team, RIKEN Center for Advanced Intelligence Project. His research interests include computional learning theory, machine learning, and computer science related to library.

Eiji Takimoto received Dr.Eng. degree from Tohoku University in 1991. Currently, he is a professor at Department of Informatics, Information Science and Electrical Engineering in Kyushu University. His research interests include computational complexity, computational learning theory, and online learning.