Technical Reports on Mathematical and Computing Sciences: TR-C149
title: An application of a random sampling technique to primal-form maximal-margin classifiers
author: Jose Balcázar[1], Yang Dai[2], and Osamu Watanabe[2]
affiliation:
1. Departament de Llenguatges i Sistemes Informatics, Univ. Politecnica de Catalunya
   Campus Nord, Jordi Girona Salgado 1-3, 08034 Barcelona, Spain
2. Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology
   Meguro-ku Ookayama, Tokyo 152-8552

email: watanabe@is.titech.ac.jp
acknowledgments to financial supports:
The first author is supported by EU ESPRIT IST-1999-14186 (ALCOM-FT), EU EP27150 (Neurocolt II), Spanish Government PB98-0937-C04-04, and CIRIT 1997SGR-00366.
The third author is supported by a Grant-in-Aid for Scientific Research on Priority Areas "Discovery Science" from the Ministry of Education, Science, Sports and Culture of Japan.

Abstract. Random sampling techniques have been developed in for geometric/combinatorial optimization problems; see, e.g., [Cla88, Cla95, AS93, GW99]. In this note, we apply one of these techniques for obtaining (hopefully) efficient support vector machine training algorithm. In particular, we propose one way to find "outliers" by using the sampling technique.

1. Preliminaries: the Support Vector Machine and the Sampling Lemma

The *support vector machine* (SVM in short) [CV95] is a machinery for two-group classification problems. Training SVM for a given set of labeled examples can be regarded as the problem of computing a hyperplane separating positive and negative examples with the largest margin.

Suppose that we are given a set of m examples $\boldsymbol{x}_i$, $1 \leq i \leq m$, in some n dimension space, say $\mathbb{R}^n$. Each example $\boldsymbol{x}_i$ is labeled by $y_i \in \{1, -1\}$ denoting the classification of the example. (Throughout this note, we use n and m to denote the dimension of the example space and the number of examples respectively. Also we use i for indexing examples (and their labels), and $\boldsymbol{x}_i$ and y_i to denote the ith example and its label respectively. The range of i is always $\{1, ..., m\}$.)

The SVM training problem we will discuss in this note is essentially to solve the following optimization problem. (This is the separable case. The nonseparable case, which is our main topic, will be discussed later.)

Max Margin (P1)

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 - (\theta_+ - \theta_-) \quad \text{w.r.t. } \boldsymbol{w} = (w_1, ..., w_n),\ \theta_+, \text{ and } \theta_-, \\ \text{s.t.} \quad & \boldsymbol{w} \cdot \boldsymbol{x}_i \ \geq \ \theta_+ \quad \text{if } y_i = 1, \\ & \boldsymbol{w} \cdot \boldsymbol{x}_i \ \leq \ \theta_- \quad \text{if } y_i = -1. \end{aligned}$$

Here we follow [BB00] and use their formulation. But this problem can be restated by using a single threshold parameter as given in the original paper [CV95]. That is, the number of essential parameters is $n + 1$; hence, the combinatorial dimension of this optimization problem is $n + 1$.

We can solve this optimization problem by using a standard general quadratic programming algorithm. In most applications, however, the number m of examples is much larger than the dimension n (in other words, much more constraints than variables), and we may be able to use a randomized sampling technique for improving efficiency.

We first explain the abstract framework given by Gärtner and Welzl [GW99] and prove the *Sampling Lemma* — a key lemma for randomized sampling techniques. Then we show an efficient way to solve the above optimization problem if m is much larger than n.

Randomized sampling techniques, particularly, the Sampling Lemma, is applicable for many "LP-type" problems. Here we use $(\mathcal{D}, \phi)$ to denote an abstract LP-type problem, where $\mathcal{D}$ is a set of elements and ϕ is a function mapping any $\mathcal{R} \subseteq \mathcal{D}$ to some value space. For formulating the problem (P1), for example, we set $\mathcal{D}$ the set of all constraints and define ϕ as a mapping from a given $\mathcal{R} \subseteq \mathcal{D}$ to the min. value of (P1) when constraints are restricted to $\mathcal{R}$. As a LP-type problem, we require $(\mathcal{D}, \phi)$ to satisfy certain conditions, but here we omit explaining them because our example case naturally satisfies these consitions.

For any $\mathcal{R} \subseteq \mathcal{D}$, a *basis* of $\mathcal{R}$ is an inclusion-minimal subset $\mathcal{B}$ of $\mathcal{R}$ such that $\phi(\mathcal{B}) = \phi(\mathcal{R})$. The *combinatorial dimension* of $(\mathcal{D}, \phi)$ is the size of the largest basis of $\mathcal{D}$. We will use δ to denote the combinatorial dimension. Note that the combinatorial dimension of the problem (P1) is $n + 1$.

Consider any set $\mathcal{U}$ of U elements in $\mathcal{D}$. $\mathcal{U}$ may be a *multiple set*, a set containing some element of $\mathcal{D}$ more than once. (We may naturally extend ϕ for multiple sets.) A *violator* of $\mathcal{R}$ is an element e of $\mathcal{D}$ such that $\phi(\mathcal{R} \cup \{e\}) \neq \phi(\mathcal{R})$. An element e of $\mathcal{R}$ is *extreme* (or, simply called an *extremer*) if $\phi(\mathcal{R} - \{e\}) \neq \phi(\mathcal{R})$. Though obvious, the following relation is important for our discussion.

$$e \text{ violates } \mathcal{R} \;\Leftrightarrow\; e \text{ is extreme in } \mathcal{R} \cup \{e\}. \tag{1}$$

Define $V(\mathcal{R})$ and $X(\mathcal{R})$ to be the number of violators and extremers of $\mathcal{R}$ respectively. Again though immediate from the definition, the following bound is important.

$$X(\mathcal{R}) \;\leq\; \delta \;(=\; \text{the combinatorial dimension of } (\mathcal{D}, \phi)). \tag{2}$$

For a given r, $0 \leq r < U$, we consider the situation where a set $\mathcal{R}$ of size r is chosen uniformly at random from $\mathcal{U}$; more precisely speaking, we select $\mathcal{R}$ uniformly at random from $\binom{U}{r}$ subsets of $\mathcal{U}$ with r elements. We consider the following expected values v_r and x_r.

$$v_r \stackrel{\text{def}}{=} \mathrm{Exp}_{\mathcal{R}}(V(\mathcal{R})), \quad \text{and} \quad x_r \stackrel{\text{def}}{=} \mathrm{Exp}_{\mathcal{R}}(X(\mathcal{R})).$$

Now we state the Sampling Lemma and its proof.

Lemma 1.1.

$$(r+1)v_r \;=\; x_{r+1}(U - r).$$

Proof. The relation is derived as follows. (Here we use $\binom{\mathcal{U}}{r}$ to denote the set of all r-element subsets of $\mathcal{U}$.)

$$\begin{aligned}\binom{U}{r} v_r &= \sum_{\mathcal{R}\in\binom{\mathcal{U}}{r}} \sum_{e\in\mathcal{U}-\mathcal{R}} [\ e \text{ violates } R\ ] \\ &= \sum_{\mathcal{R}\in\binom{\mathcal{U}}{r}} \sum_{e\in\mathcal{U}-\mathcal{R}} [\ e \text{ is extreme in } \mathcal{R}\cup\{e\}\ ] \\ &= \sum_{\mathcal{Q}\in\binom{\mathcal{U}}{r+1}} \sum_{e\in\mathcal{Q}} [\ e \text{ is extreme in } \mathcal{Q}\ ] \ = \ x_{r+1}\binom{U}{r+1}.\end{aligned}$$

□

With the bound (2), we can derive the following bound.

$$v_r \ \le \ \frac{U-r}{r+1}\cdot\delta. \tag{3}$$

The idea of this Sampling Lemma can be found in the paper by Clarkson [Cla88], where a randomized algorithm for linear programming has been proposed. Also a similar idea has been used [AS93] to design an efficient randomized algorithm for quadratic programming. In fact, by using the above bound, one can design efficient randomized algorithms for both linear programming and quadratic programming. Here we take essentially the same approach to solve (P1).

For stating the algorithm for the problem (P1), let us prepare some symbols and notions. As mentioned above, $\mathcal{D}$ corresponds to the set of constraints of (P1); but we can identify the ith constraint with the corresponding example $\boldsymbol{x}_i$. Thus, in the following, we simply use $\mathcal{X}$, the set of examples, for $\mathcal{D}$. Let $\mathcal{U}$ denote a multiple set consisting of elements from $\mathcal{X}$. For any subset $\mathcal{R}$ of $\mathcal{U}$, a *solution* of (P1) for $\mathcal{R}$ is a triple $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*)$ that (i) satisfies the constraints (corresponding to $\boldsymbol{x}_i$) in $\mathcal{R}$, and (ii) minimizes $\|\boldsymbol{w}\|^2/2 - (\theta_+ - \theta_-)$. We say that $\boldsymbol{x}_i \in \mathcal{X}$ *violates* $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*)$ if the constraint corresponding to $\boldsymbol{x}_i$ is not satisfied with $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*)$. (More precisely, $\boldsymbol{x}_i$ is a violator of $\mathcal{R}$ if the constraint corresponding to $\boldsymbol{x}_i$ is not satisfied with *some* solution of $\mathcal{R}$.) In the following algorithm, we say "duplicate $\boldsymbol{x}_i$", by which we mean to make a copy for each $\boldsymbol{x}_i$ in $\mathcal{U}$ (there may be more than one $\boldsymbol{x}_i$'s in $\mathcal{U}$) and put them into $\mathcal{U}$. For any element $\boldsymbol{x}_i \in \mathcal{X}$, let $u(\boldsymbol{x}_i)$ denote the *weight* of $\boldsymbol{x}_i$ (in $\mathcal{U}$) which is simply the total number of $\boldsymbol{x}_i$ in $\mathcal{U}$. (In order to distinguish the "number of elements" in an ordinary set, we use "weight" for referring the number of elements in a multiset. But we use the notation such as $|\mathcal{X}|$, $|\mathcal{U}|$, etc., to denote the number of elements in a set for both multiple and ordinary sets.)

Theorem 1.2. The average number of iterations executed in the OptMargin algorithm is bounded by $6\delta \ln|\mathcal{X}| = O(n \ln m)$. (We use $|\mathcal{X}|$ to denote the number of elements in $\mathcal{X}$. Recall that $|\mathcal{X}| = m$ and $\delta = n + 1$.)

Proof. We say a repeat-iteration is *successful* if the if-condition holds in the iteration. We first bound the number of successful iterations.

procedure OptMargin
  $\mathcal{U} \leftarrow \mathcal{X}$; % $|\mathcal{U}| = |\mathcal{X}| = m$.
  $r \leftarrow 6\delta^2$; % $\delta = n + 1$.
  repeat
    $\mathcal{R} \leftarrow$ an r-element subset of $\mathcal{U}$ chosen uniformly at random;
    $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*)$ is a solution of (P1) for $\mathcal{R}$;
    $\mathcal{V} \leftarrow$ the set of violators in $\mathcal{U}$ of the solution;
    if $u(\mathcal{V}) \leq u(\mathcal{U})/3\delta$ **then** duplicate $\boldsymbol{x}_i$ for all $\boldsymbol{x}_i \in \mathcal{V}$;
  until $\mathcal{V} = \emptyset$;
  return the last solution;
end-procedure.

Figure 1: Randomized SVM Training Algorithm

Notice first that $\mathcal{U}$ is a multiset and the weight function u just counts the number of appearences of a given element in $\mathcal{U}$. Thus, $u(\mathcal{V})$ is the same as $V(\mathcal{R})$, and similarly, $u(\mathcal{U}) = U$ ($\stackrel{\text{def}}{=} |\mathcal{U}|$, the number of elements in $\mathcal{U}$). Hence, at each successful iteration, we have $V(\mathcal{R}) \leq U/3\delta$. This means that by duplicating violators, U increases at most $U/3\delta$. Thus, after t successful iterations, we have $U \leq m(1 + 1/3\delta)^t$.

Let $\mathcal{X}_0$ be the set of extremers of $\mathcal{X}$. Note that if all extremers of $\mathcal{X}$ are chosen to $\mathcal{R}$, i.e., $\mathcal{X}_0 \subseteq \mathcal{R}$, then there should be no violator for $\mathcal{R}$. Thus, at each successful iteration (if it is not the end) some $\boldsymbol{x}_i$ of $\mathcal{X}_0$ must not be in $\mathcal{R}$, which in turn is a violator of $\mathcal{R}$. Hence, $u(\boldsymbol{x}_i)$ gets doubled. Since $|\mathcal{X}_0| \leq \delta$, there is some $\boldsymbol{x}_i$ in $\mathcal{X}_0$ that gets doubled at least once every δ successful iterations. Therefore, after t successful iterations, $u(\boldsymbol{x}_i) \geq 2^{t/\delta}$.

We have the following upper and lower bounds for $u(\mathcal{U})$.

$$2^{t/\delta} \;\leq\; u(\mathcal{U}) \;\leq\; m(1 + 1/3\delta)^t.$$

This implies that $t < 3\delta \ln m$ (if the repeat-condition does not hold after t successful iterations). That is, the algorithm terminates within $3\delta \ln m$ successful iterations.

Now by our choice of r, it follows from the bound (3) that the expectation of $u(\mathcal{V})$ (i.e., v_r) is bounded by $U/6\delta$. Thus, the probability that the if-condition is satisfied is at least $1/2$. This implies that the expected number of iterations is at most twice as large as the number of successful iterations. Therefore, the algorithm terminates *on average* within $2 \cdot 3\delta \ln m$ steps. □

Thus, while our randomized OptMargin algorithm needs to solve (P1) for about $6n \ln m$ times on average, the number of constraints needed to consider at each time is about $6n^2$. Hence, if n is much smaller than m, say n is about $m^{1/3}$, then this algorithm is faster than solving (P1) directly.

2. The Nonseparable Case with Outliers and a Geometrical View

For the separable case, the above randomized approach seems to help us to reduce the size of the optimization problem we need to solve for training SVM. On the other hand, the important

feature of SVM is that it is also applicable for the nonseparable case. More precisely speaking, the nonseparable case includes two subcases: (i) the case where the hyperplane classifier is too weak for classifying given examples, and (ii) the case where there are some erroneous examples, namely outliers. The first subcase is solved by the SVM approach by mapping examples into a much higher dimension space. The second subcase is solved by relaxing constraints by introducing slack variables or "soft margin error". In this note, we discuss a way to handle the second subcase; that is, the nonseparable case with outliers.

First we generalize the problem (P1) and state the soft margin hyperplane separation problem.

Max Soft Margin (P2)

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 - (\theta_+ - \theta_-) + D \cdot \sum_i \xi_i \\ & \text{w.r.t. } \boldsymbol{w} = (w_1, ..., w_n),\ \theta_+,\ \theta_-, \text{ and } \xi_1, ..., \xi_m \\ \text{s.t.} \quad & \boldsymbol{w} \cdot \boldsymbol{x}_i \ \geq \ \theta_+ - \xi_i \quad \text{if } y_i = 1, \\ & \boldsymbol{w} \cdot \boldsymbol{x}_i \ \leq \ \theta_- + \xi_i \quad \text{if } y_i = -1, \\ & \xi_i \ \geq \ 0. \end{aligned}$$

Here $D < 1$ is a parameter that determines the degree of influence from outliers. Note that D should be fixed in advance; that is, D is a constant throughout the training process.

This problem is again a quadratic programming with linear constraints; thus, it is possible to use our random sampling technique. Unfortunately, however, the technique is no longer useful because we now have $n + m + 2$ variables including $\xi_1, ..., \xi_m$. In particular, the number of variables increases proportionally to the number of examples, which is not good for solving quadratic programming in general.

Like the other optimization problems, this problem can be viewed in several different ways. The purpose of this note is to use some view and propose a variation of the sampling technique that works effectively for the problem (P2). In particular, we will make use of an intuitive geometric interpretation to (P2) that has been given by Bennett and Bredensteiner [BB00].

In what follows, we will introduce one reformulation of (P2) that could be more suitable for the sampling technique.

Bennett and Bredensteiner [BB00] proved that (P2) is equivalent to the following problem (P3). (The problem (P3) is the Wolfe dual of (P2).)

Reduced Convex Hull (P3)

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}\left\|\sum_i y_i s_i \boldsymbol{x}_i\right\|^2 \quad \text{w.r.t. } s_1, ..., s_m \\ \text{s.t.} \quad & \sum_{i:\, y_i = 1} s_i \ = \ 1, \quad \sum_{i:\, y_i = -1} s_i \ = \ 1, \quad 0 \leq s_i \leq D. \end{aligned}$$

Note that

$$\left\|\sum_i y_i s_i \boldsymbol{x}_i\right\|^2 \ = \ \left\|\sum_{i:\, y_i = 1} s_i \boldsymbol{x}_i - \sum_{i:\, y_i = -1} s_i \boldsymbol{x}_i\right\|^2.$$

That is, this is the distance between two points in the convex hull of positive and negative examples. In the separable case, it is the distance between two closest points in two convex hulls. On the other hand, in the nonseparable case, we give some restriction to the influence of each example; each example cannot contribute to the closest point more than D.

As they mentioned, the meaning of D is intuitively explained by considering its inverse $k = 1/D$. (Here we assume that $1/D$ is an integer. Throughout this note, we use k to denote this constant.) Instead of the original convex hulls, we consider the convex hulls of points composed from k examples. Then resulting convex hulls are reduced ones and they may be separable by some hyperplane; in the extreme case where $k = m_+$ (where m_+ is the number of positive examples), the reduced convex hull for positive examples consists of only one point.

More formally, we can reformulate (P3) as follows. Let $\mathcal{Z}$ be the set of *composed examples* $\boldsymbol{z}_I$ that is defined by

$$\boldsymbol{z}_I \;=\; \frac{\boldsymbol{x}_{i_1} + \boldsymbol{x}_{i_2} + \cdots + \boldsymbol{x}_{i_k}}{k},$$

with some k distinct elements $\boldsymbol{x}_{i_1}$, $\boldsymbol{x}_{i_2}$, ..., $\boldsymbol{x}_{i_k}$ of $\mathcal{X}$ with the same label (i.e., $y_{i_1} = y_{i_2} = \cdots = y_{i_k}$). The label y_I of the composed example $\boldsymbol{z}_I$ inherits its members'. Throughout this note, we use I for indexing elements of $\mathcal{Z}$ and their labels. The range of I is $\{1, ..., M\}$, where $M \stackrel{\text{def}}{=} |\mathcal{Z}|$. Note that $M \leq \binom{m}{k}$. For each $\boldsymbol{z}_I$, we use z_I to denote the set of original examples from which $\boldsymbol{z}_I$ is composed.

Then (P3) is equivalent to the following (P4).

Convex Hull of Reduced Points (P4)

$$\begin{array}{lll} \text{min.} & \dfrac{1}{2}\left\|\displaystyle\sum_I y_I s_I \boldsymbol{z}_I\right\|^2 & \text{w.r.t. } s_1, ..., s_M \\ \text{s.t.} & \displaystyle\sum_{I:\, y_I = 1} s_I \;=\; 1, \quad \sum_{I:\, y_I = -1} s_I \;=\; 1, & 0 \leq s_I \leq 1. \end{array}$$

Finally we consider the Wolfe primal of this problem. Then we came back to our favorite formulation!

Max Margin for Composed Examples (P5)

$$\begin{array}{lll} \text{min.} & \dfrac{1}{2}\|\boldsymbol{w}\|^2 - (\eta_+ - \eta_-) & \text{w.r.t. } \boldsymbol{w} = (w_1, ..., w_n),\ \eta_+, \text{ and } \eta_- \\ \text{s.t.} & \boldsymbol{w} \cdot \boldsymbol{z}_I \;\geq\; \eta_+ & \text{if } y_I = 1, \\ & \boldsymbol{w} \cdot \boldsymbol{z}_I \;\leq\; \eta_- & \text{if } y_I = -1. \end{array}$$

Note that the combinatorial dimension of (P5) is $n + 1$, the same as that of (P1). The difference is that we have now $M = O(m^k)$ constraints, which is quite large. But this situation is suitable for the sampling technique. By using our OptMargin algorithm for (P5), we can reduce the size of constraints; that is, we only have to solve (P5) with $O(n^2)$ constraints, and the expected number of iterations is $O(n \ln M) = O(kn \ln m)$.

Unfortunately, however, there are serious problems. For finding violators, we have to go through $\mathcal{Z}$, which needs $O(M)$ steps. Also for modifying weights and choosing composed examples randomly according to their weights, we need at least $O(M)$ steps. But $M = O(m^k)$ is too large. Thus, we have to consider some more efficient way.

3. A Modified Random Sampling Algorithm

Here we give one modification of the random sampling technique, and propose an alternative algorithm to OptMargin. We belive that the algorithm works efficiently for composed examples. Unfortunately, however, some hypothesis is necessary for justifying it.

Before stating our algorithm and its analysis, let us first examine solutions to the problems (P2) and (P5). Throughout this section, we fix a given example set $\mathcal{X}$ and, hence, the set $\mathcal{Z}$ of composed examples. Let $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*)$ and $(\boldsymbol{w}^*, \eta_+^*, \eta_-^*)$ be the solutions of (P2) for $\mathcal{X}$ and (P5) for $\mathcal{Z}$ respectively. Note that two solutions share the same $\boldsymbol{w}^*$; this is because (P2) and (P5) are essentially equivalent problems [BB00].

The notion of "outlier" is now formally defined. An example is called an *outlier* if it is misclassified by the solution of (P2). Let $\mathcal{X}_{\mathrm{err},+}$ and $\mathcal{X}_{\mathrm{err},-}$ denote the sets of positive/negative outliers. That is, $\boldsymbol{x}_i$ belongs to $\mathcal{X}_{\mathrm{err},+}$ (resp., $\mathcal{X}_{\mathrm{err},-}$) if and only if $y_i = 1$ and $\boldsymbol{w}^* \cdot \boldsymbol{x}_i < \theta_+^*$ (resp., $y_i = -1$ and $\boldsymbol{w}^* \cdot \boldsymbol{x}_i > \theta_-^*$). We use $k_{\mathrm{err},+}$ and $k_{\mathrm{err},-}$ to denote $|\mathcal{X}_{\mathrm{err},+}|$ and $|\mathcal{X}_{\mathrm{err},-}|$ respectively. Note that we assume that our constant k is larger than both $k_{\mathrm{err},+}$ and $k_{\mathrm{err},-}$. Let $\mathcal{X}_{\mathrm{err}} = \mathcal{X}_{\mathrm{err},+} \cup \mathcal{X}_{\mathrm{err},-}$. In our algorithm, we would like to identify this set $\mathcal{X}_{\mathrm{err}}$.

The problem (P5) is regarded as the LP-type problem $(\mathcal{D}, \phi)$, where the correspondence is the same as (P1) except that $\mathcal{Z}$ is used as $\mathcal{D}$ here. Let $\mathcal{Z}_0$ be the basis of $\mathcal{Z}$. (In order to simplify our discussion, we assume nondegeneracy throughout the following discussion.) Note that every element of the basis is extreme in $\mathcal{Z}$. Hence, we call elements of $\mathcal{Z}_0$ *final extremers*. By definition, the solution of (P5) for $\mathcal{Z}$ is defined by the constraints corresponding to these final extremers.

By analyzing the Karush-Kuhn-Tucker (in short, KKT) condition for (P2), we can show the following facts.

Lemma 3.1. Let $\boldsymbol{z}_I$ be any positive final extremer, i.e., an element of $\mathcal{Z}_0$ such that $y_I = 1$. Then the following properties hold. (Though the lemma is stated only for the positive case, i.e., the case $y_I = 1$, the corresponding properties hold for the negative case $y_I = -1$.)

(a) $\boldsymbol{w}^* \cdot \boldsymbol{z}_I = \eta_+^*$.

(b) $\mathcal{X}_{\mathrm{err},+} \subseteq z_I$.

(c) For every $\boldsymbol{x}_i \in z_I$, if $\boldsymbol{x}_i \notin \mathcal{X}_{\mathrm{err},+}$, then we have $\boldsymbol{w}^* \cdot \boldsymbol{x}_i = \theta_+^*$.

Proof. (a) Since $\mathcal{Z}_0$ is the set of final extremers, (P5) can be solved only with the constraints corresponding to elements in $\mathcal{Z}_0$. Suppose that $\boldsymbol{w}^* \cdot \boldsymbol{z}_J > \eta_+^*$ (resp., $\boldsymbol{w}^* \cdot \boldsymbol{z}_J < \eta_-^*$) for some positive (resp., negative) $\boldsymbol{z}_J \in \mathcal{Z}_0$ including $\boldsymbol{z}_I$ of the lemma. Let $\mathcal{Z}'$ be the set of such $\boldsymbol{z}_J$'s of $\mathcal{Z}_0$. If $\mathcal{Z}'$ indeed contained all positive examples in $\mathcal{Z}_0$, then we could set θ_+ with $\theta_+^* - \epsilon$ for some $\epsilon > 0$ and still satisfy all the constraints, which contradicts the optimality of the solution. Hence, we may assume that $\mathcal{Z}_0 - \mathcal{Z}'$ still has some positive example. Then it is well known (see, e.g., [DB95]) that a local optimal solution to the problem (P5) with the constraints corresponding to elements in $\mathcal{Z}_0$ is also locally optimal to the problem (P5) with the constraints corresponding to only elements in $\mathcal{Z}_0 - \mathcal{Z}'$. Furthermore, since (P5) is a convex programming, a local optimal solution is globally optimal. Thus, the original problem (P5) is solved with the

constrains corresponding to elements in $\mathcal{Z}_0 - \mathcal{Z}'$. This contradicts our assumption that $\mathcal{Z}_0$ is the set of final extremers.

(b) Consider the KKT-point $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \boldsymbol{\xi}^*, \boldsymbol{s}^*, \boldsymbol{u}^*)$ of (P2). Then the point must satisfy the following so called KKT-condition. (Below we use i to denote indices of examples, and let P and N respectively denote indices i of examples such that $y_i = 1$ and $y_i = 0$. We use $\boldsymbol{e}$ to denote the vector with 1 at every entry.)

$$\begin{aligned}
\boldsymbol{w}^* - \sum_{i\in P} s_i \boldsymbol{x}_i + \sum_{i\in N} s_i^* \boldsymbol{x}_i &= 0, & D\boldsymbol{e} - \boldsymbol{s}^* - \boldsymbol{u}^* &= 0,\\
-1 + \sum_{i\in P} s_i^* &= 0, & -1 + \sum_{i\in N} s_i^* &= 0,\\
\forall i \in P\ [\ s_i^*(\boldsymbol{w}^* \cdot \boldsymbol{x}_i - \theta_+^* + \xi_i^*) &= 0\ ], & \forall i \in N\ [\ s_i^*(\boldsymbol{w}^* \cdot \boldsymbol{x}_i - \theta_-^* - \xi_i^*) &= 0\ ],\\
\boldsymbol{u}^* \cdot \boldsymbol{\xi}^* &= 0 \text{ (which means } (D\boldsymbol{e} - \boldsymbol{s}^*) \cdot \boldsymbol{\xi}^* = 0), \text{ and } & \boldsymbol{\xi}^*, \boldsymbol{u}^*, \boldsymbol{s}^* &\geq 0.
\end{aligned}$$

Note that $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \boldsymbol{\xi}^*)$ is an optimal solution of (P2), since (P2) is a convex minimization problem. From these requirements, we have the following relation. (Note that the condition $\boldsymbol{s}^* \leq D\boldsymbol{e}$ below is derived from the requirements $D\boldsymbol{e} - \boldsymbol{s}^* - \boldsymbol{u}^* = 0$ and $\boldsymbol{u}^* \geq 0$.)

$$\begin{aligned}
\boldsymbol{w}^* &= \sum_{i\in P} s_i^* \boldsymbol{x}_i - \sum_{i\in N} s_i^* \boldsymbol{x}_i,\\
\sum_{i\in P} s_i^* &= 1, \quad \sum_{i\in N} s_i^* = 1, \text{ and } 0 \leq \boldsymbol{s}^* \leq D\boldsymbol{e}.
\end{aligned}$$

In fact, $\boldsymbol{s}^*$ is exactly the optimal solution of (P3).

Here by the equivalence of (P4) and (P5), we see that the final extremers are exactly points contributing to the solution of (P4). That is, we have $\boldsymbol{z}_I \in \mathcal{Z}_0$ if and only if $s_I^* > 0$, where s_I^* is the Ith element of the solution of (P4). Furthermore, it follows the equivalence between (P3) and (P4), for any i, we have

$$\frac{1}{k} \cdot \sum_{I:\boldsymbol{x}_i \in z_I} s_I^* = s_i^*. \tag{4}$$

Recall that each z_I is defined as the center of k examples of $\mathcal{X}$. Hence, to show that every $\boldsymbol{x}_i \in \mathcal{X}_{\mathrm{err},+}$ appears in all positive final extremers, it suffices to show that $s_i^* = 1/k$ for every $\boldsymbol{x}_i \in \mathcal{X}_{\mathrm{err},+}$, which follows from the following argument. For any $\boldsymbol{x}_i \in \mathcal{X}_{\mathrm{err},+}$, since $\xi_i^* > 0$, it follows from the requirements $(D\boldsymbol{e} - \boldsymbol{s}^*) \cdot \boldsymbol{\xi}^* = 0$ and $D\boldsymbol{e} - \boldsymbol{s}^* \geq 0$ that $D - s_i^* = 0$; that is, $s_i^* = 1/k$ for any $\boldsymbol{x}_i \in \mathcal{X}_{\mathrm{err},+}$.

(c) Consider any index i in P such that $\boldsymbol{x}_i$ appears in some of the final extremer $\boldsymbol{z}_I \in \mathcal{Z}_0$. Since $s_I^* > 0$, we can show that $s_i^* > 0$ by using the equation (4). Hence, from the requirement $s_i(\boldsymbol{w}^* \cdot \boldsymbol{x}_i - \theta_+^* + \xi_i^*) = 0$, we have

$$\boldsymbol{w}^* \cdot \boldsymbol{x}_i - \theta_+^* + \xi_i^* = 0.$$

Thus, if $\boldsymbol{x}_i \notin \mathcal{X}_{\mathrm{err}}$, i.e., it is not an outlier or $\xi_i^* = 0$, then we have $\boldsymbol{w}^* \cdot \boldsymbol{x}_i = \theta_+^*$. □

Therefore, when the final extremers are obtained, the set of outliers can be obtained as $\cap_{\boldsymbol{z}_I \in \mathcal{Z}_0} z_I$, the set of examples that appear in all final extremers.

Let us give some intuitive interpretation to the facts given in this lemma. (Again we only consider, for the simplicity, the positive examples.) The fact (a) of the lemma shows that all final extremers are located on some hyperplane whose distance from the base hyperplane $\boldsymbol{w}^* \cdot \boldsymbol{z} = 0$ is η^*_+. On the other hand, the fact (c) states that all original *normal* examples (i.e., examples not in $\mathcal{X}_{\mathrm{err},+}$) are located again on some hyperplane whose distance from the base hyperplane is $\theta^*_+ > \eta^*_+$. Here consider the point $\boldsymbol{v}_+ \stackrel{\mathrm{def}}{=} \left(\sum_{\boldsymbol{x}_i \in \mathcal{X}_{\mathrm{err},+}} \boldsymbol{x}_i\right) / k_{\mathrm{err},+}$, i.e., the center of positive outliers, and define $\mu^*_+ = \boldsymbol{w}^* \cdot \boldsymbol{v}_+$. Then we have $\theta^*_+ \geq \eta^*_+ \geq \mu^*_+$; that is, the hyperplane defined by the final extremers is located between the one having all normal examples in the final extremers and the one having the center $\boldsymbol{v}_+$ of outliers. More specifically, since every final extremer is composed from all $k_{\mathrm{err},+}$ positive outliers and $k - k_{\mathrm{err},+}$ normal examples, we have

$$\theta^*_+ - \eta^*_+ \ : \ \eta^*_+ - \mu^*_+ \ = \ k - k_{\mathrm{err},+} \ : \ k_{\mathrm{err},+}.$$

Next we consider partial solutions of (P5). We would like to solve (P5) by using the random sampling technique. That is, choose some small subset $\mathcal{R}$ of $\mathcal{Z}$ randomly according to current weight, and solve (P5) for $\mathcal{R}$. Thus, let us examine partial solutions obtained by solving (P5) for such a subset $\mathcal{R}$ of $\mathcal{Z}$.

For any set $\mathcal{R}$ of composed examples in $\mathcal{Z}$, let $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$ be the solution of (P5) for $\mathcal{R}$. Similar to the above, we consider $\widetilde{\mathcal{Z}}_0$ to be the set of extremers of $\mathcal{R}$ w.r.t. the solution $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$. On the other hand, we define here $\widetilde{\mathcal{X}}$ to be the set of original examples appearing in some extremers in $\widetilde{\mathcal{Z}}_0$.

As before, we will discuss about only positive composed/orignal examples. Let $\widetilde{\mathcal{Z}}_{0,+}$ be the set of posivite extremers. Different from the case where all composed examples are examined to solve (P5), here we cannot expect, for example, that all extremers in $\widetilde{\mathcal{Z}}_{0,+}$ share the same set of misclassified examples. Thus, instead of sets like $\mathcal{X}_{\mathrm{err},+}$, we consider a subset $\widetilde{\mathcal{X}}'_+$ of the following set $\widetilde{\mathcal{X}}_+$. (It may be the case that $\widetilde{\mathcal{X}}_+$ is empty.)

$$\widetilde{\mathcal{X}}_+ \ = \ \text{the set of positve examples appearing in all estremers in } \widetilde{\mathcal{Z}}_{0,+}.$$

Intuitively, we want to discuss by using the set $\widetilde{\mathcal{X}}'_+$ of "misclassified" examples appearing in all positive extremers. But such a set cannot be defined at this point because no threshold corresponding to θ^*_+ has been defined. Thus, for a while, let us consider any subset $\widetilde{\mathcal{X}}'_+$ of $\widetilde{\mathcal{X}}_+$. Let $\widetilde{k}'_+ = |\widetilde{\mathcal{X}}'_+|$, and define $\boldsymbol{v}_+ = \left(\sum_{\boldsymbol{x}_i \in \widetilde{\mathcal{X}}'_+} \boldsymbol{x}_i\right) / \widetilde{k}'_+$. Also for each $\boldsymbol{z}_I \in \widetilde{\mathcal{Z}}_0$, we define a point $\boldsymbol{v}_I$ that is the center of all original examples in $z_I - \widetilde{\mathcal{X}}'_+$. That is,

$$\boldsymbol{v}_I \stackrel{\mathrm{def}}{=} \frac{\sum_{\boldsymbol{x}_i \in z_I - \widetilde{\mathcal{X}}'_+} \boldsymbol{x}_i}{k - \widetilde{k}'_+}.$$

Then we can prove the following fact that corresponds to Lemma 3.1 and that is proved similarly. (We omit the proof.)

Lemma 3.2. For any subset $\mathcal{R}$ of $\mathcal{Z}$, we use the symbols defined as above. There exists some $\widetilde{\theta}'_+$ such that for any extremer $\boldsymbol{z}_I$ in $\widetilde{\mathcal{Z}}_{0,+}$, we have $\widetilde{\boldsymbol{w}} \cdot \boldsymbol{v}_I = \widetilde{\theta}'_+$. (The corresponding fact also holds for the nagative case.)

Now for our $\widetilde{\mathcal{X}}_{\mathrm{err},+}$, we use a subset $\widetilde{\mathcal{X}}'_+$ of $\widetilde{\mathcal{X}}_+$ satisfying the following for any $\boldsymbol{x}_i \in \widetilde{\mathcal{X}}_+$.

$$\boldsymbol{x}_i \in \widetilde{\mathcal{X}}'_+ \;\Leftrightarrow\; \widetilde{\boldsymbol{w}} \cdot \boldsymbol{x}_i \;<\; \widetilde{\theta}'_+,$$

where $\widetilde{\theta}'_+$, which we denote $\widetilde{\theta}_{\mathrm{err},+}$, is the threshold given in Lemma 3.2 for $\widetilde{\mathcal{X}}'_+$. Such a set (while it could be empty) is well-defined. (In the case that $\widetilde{\mathcal{X}}_{\mathrm{err},+}$ is empty, we define $\widetilde{\theta}_{\mathrm{err},+} = \widetilde{\eta}_+$.)

For any original positive example $\boldsymbol{x}_i \in \mathcal{X}$, we call it a *missed example* (w.r.t. the partial solution $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$) if $\boldsymbol{x}_i \notin \widetilde{\mathcal{X}}_{\mathrm{err},+}$ and it holds that

$$\widetilde{\boldsymbol{w}} \cdot \boldsymbol{x}_i \;<\; \widetilde{\theta}_{\mathrm{err},+}. \tag{5}$$

(A negative missed example is defined similarly.) We will use such a missed example as an evidence that there exists a "violator" to $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$, which is guranteed by the following lemma.

Lemma 3.3. For any subset $\mathcal{R}$ of $\mathcal{Z}$, let $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$ be the solution of (P5) for $\mathcal{R}$. Then if there exists a missed example w.r.t. $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$, then we have some composed example in $\mathcal{Z}$ that is misclassified w.r.t. $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$. On the other hand, for any composed example $\boldsymbol{z}_I \in \mathcal{Z}$, if it is misclassified w.r.t. $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$, then z_I contains some missed example.

Proof. We consider again only the positive case. Suppose that some missed positive example $\boldsymbol{x}_i$ exists. By definition, we have $\widetilde{\boldsymbol{w}} \cdot \boldsymbol{x}_i < \widetilde{\theta}_{\mathrm{err},+}$, and there exists some extremer $\boldsymbol{z}_I \in \widetilde{\mathcal{Z}}_{0,+}$ that does not contain $\boldsymbol{x}_i$. Clearly, $\boldsymbol{z}_I$ contains some example $\boldsymbol{x}_j$ such that $\widetilde{\boldsymbol{w}} \cdot \boldsymbol{x}_j \geq \widetilde{\theta}_{\mathrm{err},+}$. Then we can easily see that a composed elements $\boldsymbol{z}_J$ consisting of $z_I - \{\boldsymbol{x}_j\} \cup \{\boldsymbol{x}_i\}$ does not satisfy the constraint $\widetilde{\boldsymbol{w}} \cdot \boldsymbol{z}_J \geq \widetilde{\eta}_+$.

For proving the second statement, note first that any "misclassified" original example $\boldsymbol{x}_i$, i.e., an example for which the inequality (5) holds, is either a missed example or an element of $\widetilde{\mathcal{X}}_{\mathrm{err},+}$. Thus, if a composed element $\boldsymbol{z}_I$ does not contain any missed example, then it cannot contain any misclassified examples other than those in $\widetilde{\mathcal{X}}_{\mathrm{err},+}$. Then it is easy to see that $\widetilde{\boldsymbol{w}} \cdot \boldsymbol{z}_I \geq \widetilde{\eta}_+$; that is, $\boldsymbol{z}_I$ is not misclassfied w.r.t. $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$. □

Now we explain the idea of our new random sampling algorithm. As before, we choose (according to some weight) a set $\mathcal{R}$ consisiting of R composed examples in $\mathcal{Z}$, and then solve (P5) for $\mathcal{R}$. In the original sampling algorithm, this sampling is repeated until no violator exists. Recall that we are regarding (P5) as a LP-type problem and that by "a violator of $\mathcal{R}$", we mean a composed example that is misclassified with the current solution $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$ of (P5) obtained for $\mathcal{R}$. Thanks to the above lemma, we do not have to go through all composed examples in order to search for a violator. A violator exists if and only if there exists some missed example w.r.t. $(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-)$. Thus, our first idea is to use the existence of missed example for the stopping condition. That is, the sampling procedure is repeated until no missed example exists.

The second idea is to use the weight of examples $\boldsymbol{x}_i$ in $\mathcal{X}$ to define the weight of composed examples. Let u_i denote the weight of the ith example $\boldsymbol{x}_i$. Then for each composed example $\boldsymbol{z}_I \in \mathcal{Z}$, its weight U_I is defined as the total of weights of all examples contained in z_I; that is, $U_I = \sum_{\boldsymbol{x}_i \in z_I} u_i$. We use symbols u and U to refer these two weight schemes. Define $u_0 = \sum_i u_i$,

procedure OptMargin&FindOutliers
$u_i \leftarrow 1$, for each i, $1 \leq i \leq m$;
$u_0 \leftarrow \sum_i u_i$; % $u_+ \stackrel{\text{def}}{=} \sum_{i:\, y_i=1} u_i$ and $u_- \stackrel{\text{def}}{=} \sum_{i:\, y_i=-1} u_i$.
$U_0 \leftarrow u_+ \binom{m_+-1}{k-1} + u_- \binom{m_--1}{k-1}$; % $U_0 = k\binom{m_+}{k} + k\binom{m_-}{k} = kM$.
$R \leftarrow 6\alpha r_0 n$; % Parameters α and r_0 are given in the discussion below
repeat % For example, we may set $r_0 = k(n+1)$ and $\alpha = 1$.
$\mathcal{R} \leftarrow$ choose R elements from $\mathcal{Z}$ without repetition according to their weights;
$(\widetilde{\boldsymbol{w}}, \widetilde{\eta}_+, \widetilde{\eta}_-) \leftarrow$ the solution of (P5) for $\mathcal{R}$;
$\widetilde{\mathcal{X}}_{\text{err}} \leftarrow$ the set of missed examples w.r.t. the above solution;
if $u(\widetilde{\mathcal{X}}_{\text{err}}) \leq u_0/(3r_0)$ **then** % $u(\widetilde{\mathcal{X}}_{\text{err}}) = \sum_{i:\, \boldsymbol{x}_i \in \widetilde{\mathcal{X}}_{\text{err}}} u_i$.
$u_i \leftarrow 2u_i$ for each $\boldsymbol{x}_i \in \widetilde{\mathcal{X}}_{\text{err}}$;
update u_0 and U_0;
until no missed example exists;
return the last solution;
% Also the set $\mathcal{X}_{\text{err}}$ of outliers is obtained from the set of final extremers.
end-procedure.

Figure 2: New Randomized SVM Training Algorithm

and $U_0 = \sum_I U_I$; that is, they are total weights. As explained below, it is computationally easy to generate each z_I with probability U_I/U_0.

Our third idea is to increase weights u_i if it is a missed example w.r.t. the current solution. More specifically, we double the weight u_i if $\boldsymbol{x}_i$ is a missed example w.r.t. the current solution for $\mathcal{R}$. Lemma 3.3 guarantees that the weight of some element of an final extremer gets doubled so long as there is some missed example. This property is crucial to estimate the number of repetitions.

Now we state our new algorithm in Figure 2. In the following, we explain some important points on this algorithm.

Random Generation of $\mathcal{R}$

We explain how to generate each $\boldsymbol{z}_I$ proportional to U_I. Again we only consider the generation of positive composed examples, and we assume that all positive examples are re-indexed as $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$. Also for simplifying our notation, we reuse m amd M to denote m_+ and M_+ respectively.

Recall that each $\boldsymbol{z}_I$ is defined as $(\boldsymbol{x}_{i_1} + \cdots + \boldsymbol{x}_{i_k})/k$, where $\boldsymbol{x}_{i_j}$ is an element of z_I. Here we assume that $i_k < i_{k-1} < \cdots < i_1$. Then each $\boldsymbol{z}_I$ is uniquely corresponds to some k-tuple $(i_k, ..., i_1)$, and we identify here the index I of $\boldsymbol{z}_I$ and this k-tuple. Let $\mathcal{I}$ be the set of all such k-tuples $(i_k, ..., i_1)$ that satisfy $1 \leq i_j \leq m$ (for each j, $1 \leq j \leq k$) and $i_k < \cdots < i_1$. Here we assume the standard lexcographic order in $\mathcal{I}$.

As stated in the above algorithm, we keep the weights $u_1, ..., u_m$ of examples in $\mathcal{X}$. By using these weights, we can calculate the total weight $U_0 = \sum_{\boldsymbol{x}_i \in \mathcal{X}} u_i$. Similarly, for each $\boldsymbol{z}_I \in \mathcal{Z}$, it is not so difficult to compute the following accumulated weight. (See Lemma 3.4 below.)

procedure generate
  $p \leftarrow$ choose one number from $\{1, ..., U_0\}$ uniformly at random;
  I_0, $I_1 \leftarrow$ the first/last element of $\mathcal{I}$;
  repeat
    $I \leftarrow$ the mid element of $\{I_0, ..., I_1\}$;
    $I' \leftarrow$ the predecessor of I;
    if $\overline{U}(I') < p \leq \overline{U}(I)$ **then**
      return $\boldsymbol{z}_I$;
    else if $\overline{U}(I') \geq p$ **then**
      $I_1 \leftarrow I'$;
    else % That is, $\overline{U}(I) < p$ holds.
      $I_0 \leftarrow I$;
  until an appropriate I is found;
end-procedure.

Figure 3: A Procedure for Generating a Composed Example

$$\overline{U}(I) \stackrel{\text{def}}{=} \sum_{J \leq I} U_J.$$

Thus, for generating $\boldsymbol{z}_I$, (i) choose p randomly from $\{1, ..., U_0\}$, and (ii) search for the smallest element I of $\mathcal{I}$ such that $\overline{U}(I) \geq p$. The second step can be done by the standard binary search in $\{1, ..., M\}$ described in Figure 3, which needs $\log M$ ($\leq k \log m$) steps.

We now state a lemma showing a way to compute $\overline{U}(I)$. First we prepare some notations. Define $\overline{V}(I)$ as follows for each $I \in \mathcal{I}$.

$$\overline{V}(I) \stackrel{\text{def}}{=} \sum_{J \geq I} U_J.$$

Then it is easy to see that (i) $U_0 = \overline{V}((1, 2, 3, ..., k))$, and (ii) $\overline{U}(I) = U_0 - \overline{V}(I) + (u_{i_k} + u_{i_{k-1}} + \cdots + u_{i_1})$ for each $I = (i_k, i_{k-1}, ..., i_1)$. Thus, it suffices to show how to compute $\overline{V}(I)$.

Consider any given $I = (i_k, i_{k-1}, ..., i_1)$ in $\mathcal{I}$. Also for any j, $1 \leq j \leq k$, we consider the prefix $I'_j = (i_j, i_{j-1}, ..., i_1)$ of I, and define the following values.

$$\begin{array}{lll} N_j & \stackrel{\text{def}}{=} & \#\text{ of } I' \text{ such that } I' \geq I'_j, \text{ and} \\ V_j & \stackrel{\text{def}}{=} & \displaystyle\sum_{I'=(i'_j,...,i'_1) \geq I'_j} (u_{i'_j} + u_{i'_j - 1} + \cdots + u_{i'_1}). \end{array}$$

Then clearly we have $\overline{V}(I) = V_k$, and our task is to compute V_k, which can be done inductively as shown in the following lemma. (The proof is easy, and thus it is omitted.)

Lemma 3.4. Consider any given $I = (i_k, i_{k-1}, ..., i_1)$ in $\mathcal{I}$, and use the symbols defined above. Then for each j, $1 \leq j \leq k$, the following relations hold. (Here we let $N_0 = 1$.)

$$N_j \;=\; N_{j-1} + \binom{m-i_j}{j}, \;\; \text{and} \;\; V_j \;=\; u_{i_j} \cdot N_{j-1} + \sum_{i_j+1 \le i \le m} u_i \cdot \binom{m-i}{j-1}.$$

Stopping Condition and Number of Successful Iterations

The correctness of our stopping condition is clear from Lemma 3.3. We estimate the number of the repeat-iterations. Here again we say that the repeat-iteration is *successful* if the if-condition holds. We give an upper bound for the number of successful iterations.

Lemma 3.5. Set $r_0 = k(n+1)$ in the algorithm. Then the number of successful iterations is at most $3k(n+1)\ln m$.

Proof. For any $t > 0$, we consider the total weight u_0 of examples in $\mathcal{X}$ after t successful iterations. As before, we can give an upper bound $u \le m(1 + 1/(3r_0))^t$. On the other hand, some missed example exists at each repeat-iteration, and from Lemma 3.3, we can indeed find it in any violator, in particular, some final extremer $\boldsymbol{z}_I \in \mathcal{Z}_0$. Thus, there must be some element $\boldsymbol{x}_i$ of $\cup_{\boldsymbol{z}_I \in \mathcal{Z}_0} z_I$ whose weight u_i gets doubled at least once every $k(n+1)$ steps. (Recall that $|\mathcal{Z}_0| \le n+1$.) Hence, we have

$$2^{t/k(n+1)} \;\le\; u_0 \;\le\; m(1 + 1/(3r_0))^t.$$

This implies, under the above choice of r_0, that $t < 3k(n+1)\ln m$. □

Remark. Here we use the fact that some element $\boldsymbol{x}_i$ of an final extremer is a missed example and hence gets doubled at each successful iteration. But it is reasonable to expect that such a missed example is, at most repeat-iterations, an outlier $\boldsymbol{x}_i \in \mathcal{X}_{\rm err}$. If we may assume so, that is, some element of $\mathcal{X}_{\rm err}$ becomes a missed example at all (or almost all) repeat-iterations, then we can set $r_0 = k$ and the above bound can be reduced to $3k \ln m$.

Our Hypothesis and the Sampling Lemma

Finally, the most important point is to estimate how often we would have successful iterations. At each repeat-iteration of our algorithm, we consider the following ratio.

$$\rho_{\rm miss} \;=\; \frac{\text{the weight of missed examples}}{u_0}.$$

Recall that the repeat-iteration is successful if this ratio is at most $1/(3r_0)$. Our hypothesis is that the ration $\rho_{\rm miss}$ is, on average, bounded by $1/(3r_0)$. Here we discuss when and for which parameter r_0, this hypothesis would hold.

For the analysis, we consider "violators" to the partial solution of (P5) obtained at each repeat-iteration. Let $\mathcal{R}$ be the set of R composed examples randomly chosen from $\mathcal{Z}$ with the probability proportional to their weights determined by U. Recall a violator of $\mathcal{R}$ is a composed

example $\boldsymbol{z}_I \in \mathcal{Z}$ that is misclassified with the obtained solution for $\mathcal{R}$. Let $\mathcal{V}$ be the set of violators, and let V_R be its weight under U. Recall also that the total weight of $\mathcal{Z}$ is U_0. Thus, by the Sampling Lemma, the ratio $\rho_{\text{vio}} \stackrel{\text{def}}{=} V_R/U_0$ is bounded as follows.

$$\text{Exp}(\rho_{\text{vio}}) \;\leq\; \frac{n+1}{R+1} \cdot \frac{U_0 - R}{U_0} \;\leq\; \frac{n}{R}.$$

Now from Lemma 3.3, we know that every violator should contain at least one missed example. On the other hand, every missed example would contribute to some violator. Hence, it seems reasonable to expect that the ratio ρ_{miss} is bounded by the ratio ρ_{vio}, or at least it holds quite often if it is not always true. Here we propose the following technical hypothesis.

(Hypothesis) $$\left(\frac{u(\widetilde{\mathcal{X}}_{\text{err}})}{u_0} \;= \right) \;\rho_{\text{miss}} \;\leq\; \rho_{\text{vio}} \;\left(= \;\frac{V_R}{U_0} \right).$$

Under this hypothesis, we have $\rho_{\text{miss}} \leq n/R$ on average; thus, by taking $R = 6\alpha r_0 n$ with $\alpha = 1$, we can show that the expected ratio ρ_{miss} is at most $1/6r_0$, which implies as before that the expected number of iterations is at most twice as the number of successful iterations.

References

[AS93] I. Adler and R. Shamir, A randomized scheme for speeding up algorithms for linear and convex programming with high constraints-to-variable ratio, *Math. Programming* 61, 39−52, 1993.

[BB00] K.P. Bennett and E.J. Bredensteiner, Duality and geometry in SVM classifiers, in *Proc. the 17th Int'l Conf. on Machine Learning* (ICML'2000), 57−64, 2000.

[Cla88] K.L. Clarkson, A Las Vegas algorithm for linear programming when the dimension is small, in *Proc. the 29th IEEE Sympos. on Foundations of Comp. Sci.* (FOCS'88), 1988. (See below for a better version.)

[Cla95] K.L. Clarkson, Las Vegas algorithms for linear and integer programming, *J.ACM* 42, 488−499, 1995.

[CV95] C. Cortes and V. Vapnik, Support-vector networks, *Machine Learning* 20, 273− 297, 1995.

[DB95] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[GW96] B. Gärtner and E. Welzl, Linear programming – Randomized and abstract frameworks, in *Proc. 13th Ann. Sympos. on Theoret. Aspects of Comput. Sci.*, Lecture Notes in Comp. Sci. 1046, 669−687, 1996.

[GW99] B. Gärtner and E. Welzl, Random sampling in geometric optimization: New insights and applications, a manuscript, 1999.