Theory Comput Syst
DOI 10.1007/s00224-007-9094-6

Provably Fast Training Algorithms for Support Vector Machines

José L. Balcázar · Yang Dai · Junichi Tanaka · Osamu Watanabe

Abstract Support Vector Machines are a family of algorithms for the analysis of data based on convex Quadratic Programming. We derive randomized algorithms for training SVMs, based on a variation of Random Sampling Techniques; these have been successfully used for similar problems. We formally prove an upper bound on the expected running time which is quasilinear with respect to the number of data points and polynomial with respect to the other parameters, i.e., the number of attributes and the inverse of a chosen soft margin parameter. [This is the combined journal version of the conference papers (Balcázar, J.L. et al. in Proceedings of 12th International Conference on Algorithmic Learning Theory (ALT'01), pp. 119–134, 2001; Balcázar, J.L. et al. in Proceedings of First IEEE International Conference on

The first and the fourth authors started this research while visiting the Centre de Recerca Matemàtica of the Institute of Catalan Studies in Barcelona.
The first author was supported by IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT), Spanish Government TIC2004-07925-C03-02, and CIRIT 2001SGR-00252.
The second author conducted this research while she was with Department of Mathematical and Computing Sciences, Tokyo Institue of Technology, and was supported by a Grant-in-Aid (C-13650444) from Japanese Goverment.
The fourth author was supported in part by a Grant-in-Aid for Scientific Research on Priority Areas "Discovery Science" 1998–2000 from Japanese Goverment.

J.L. Balcázar
Departament de Llenguatges i Sistemes Informàtics, Univ. Politècnica de Catalunya Campus Nord, Jordi Girona Salgado 1-3, 08034 Barcelona, Spain

Y. Dai
Department of Bioengineering (MC063), University Illinois at Chicago, 851 S. Morgan Str, Chicago, IL 60607-7052, USA

J. Tanaka · O. Watanabe (✉)
Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Meguro-ku Ookayama, Tokyo 152-8552, Japan
e-mail: watanabe@is.titech.ac.jp

Data Mining (ICDM'01), pp. 43–50, 2001; and Balcázar, J.L. et al. in Proceedings of SIAM Workshop in Discrete Mathematics and Data Mining, pp. 19–29, 2002).]

Keywords Support Vector Machine · SVM training algorithm · Random sampling technique · Combinatorial dimension

1 Introduction

The *Support Vector Machine* (SVM in short) is a modern mechanism for two-class classification and regression problems. Since the present form of SVM was proposed [15], SVMs have been used in various application areas, and their classification power has been investigated in depth from both experimental and theoretical points of view [16]. An important feature is that their way of working, by identifying the so-called "support vectors" among the data, offers important contributions to a number of problems related to Data Mining. The purpose of this paper is to give a fast and theoretically justified algorithm—SVM training algorithm—that finds such support vectors from a huge input data set. Our algorithm is based on a variation of Random Sampling Techniques, which have been successfully used for similar problems.

The currently employed forms of SVM can be applied to nonseparable data as well, by means of the so-called "soft margin" formulation, which will be described precisely below. In this case, outliers (i.e., erroneous data or exceptions) can be handled in the same way, while being penalized, as the other normal input data. The influence of the outliers is adjusted by the use of the so-called "soft margin parameter", which is denoted as $D \leq 1$ throughout this paper.

Algorithmically, the SVM training amounts to the identification of the solution of a convex quadratic programming (QP in short) problem. (It was proved in [30] that a similar technique is able to help choosing an appropriate kernel.) Though convex QP problems are polynomial-time solvable, they are still not so easy. According to [6] and [19], even the currently best method requires *in the worst case* $\mathrm{T}_{\mathrm{QP}}(n, m) = m^{3/2}n^2$, where, in the context of the SVM training, n is the number of attributes[1] and m is that of examples for a given data set. Thus, to scale up to really large data sets, the standard QP algorithms alone are inappropriate since their running times grow fast in m. Therefore, many algorithms and implementation techniques have been developed for training SVMs efficiently; see, e.g., [11, 12, 19, 23, 26, 33].

Among the proposed speed-up techniques, the "subset selection" [33] has been used as an effective heuristic [10], whose motivation is the same as our algorithm. Roughly speaking, the *subset selection* is a technique for the speed-up of SVM training by dividing the original QP problem into small pieces, thereby reducing the size of each QP problem. Well known variations of subset selection techniques are chunking, decomposition, and sequential minimal optimization (SMO in short); see [15, 16, 27, 29] for the details. In particular, variants of SMO have become popular because they outperform the others in several experiments. The performance of these active-subset techniques has been extensively examined, but only a few number of

[1] Precisely, n here is the number of attributes *plus* 1, but we ignore this additive difference here.

theoretical analysis have been reported. The convergence of some of such algorithms has been theoretically shown in [22, 25]; more recently, some polynomial bounds on the number of subproblem solving steps have been obtained theoretically for the SVM-light (under a certain assumption) [24] and for some variation of the subset selection technique [21]. In this paper, we propose another variation of the subset selection technique and prove that it converges on average within quasilinear subproblem solving steps and, in total, time quasilinear in m and polynomial in the other parameters. We also discuss the extension of our approach to the support vector regression, the regression problem under ϵ-insensitive loss by means of SVM.

Our algorithm seems similar to the one proposed by Pavlov et al. [28], which also involves a distribution of probability on the data points and samples like we will do. While our approach is quite similar to boosting, theirs is literally boosting, with the same constants and normalizations. However, boosting was designed with the purpose of improving the generalization error, whereas for the SVM training, the classifier to be found is always the same and the sampling only serves as an algorithmic means of faster computation; this is indeed the way of using randomness in our algorithm.

We follow a soft margin SVM formulation by Bennett and Bredensteiner [5]. With this formulation, they gave a nice geometric interpretation of the soft margin parameter $D \leq 1$. This formulation is the same as the standard soft margin SVM formulation, provided that D is small enough, more technically, small enough so that nontrivial solution exists. In other words, such a small D should be used [5] in this formulation; this criteria is in fact recommended in the context of linear programming type classification [7]. We will also show a way to determine appropriate D algorithmically. It should be also noted here that the formulation is similar to ν-SVM [31, 32]. In the ν-SVM formulation, D is simply fixed to $1/m$, while another parameter ν is introduced to give a weight to $\theta_+ - \theta_-$ (see (P5) in our Sect. 2); then the choice of ν becomes important for the ν-SVM formulation. See [5] for discussion on the relationship between the formulation of Bennett and Bredensteiner and the ν-SVM formulation.

The purpose of our research is to provide a fast and *theoretically justified* SVM training algorithm. We propose the use of the random sampling techniques that have been developed and used for combinatorial optimization problems; see, e.g., [1, 13, 20]. It is straightforward to apply some of the random sampling techniques [20] for the determination of a maximum margin separating hyperplane for the two-class classification problem, if the data set is linearly separable in the original feature space. On the other hand, the problem becomes complicated when the given data set is not linearly separable. Here the geometric interpretation of Bennett and Bredensteiner [5] is used for applying the same random sampling technique. Our algorithm solves a relatively small QP subproblem at each iteration. We prove that the expected number of these iteration steps is $O(\delta \ln m)$, where each iteration needs $O(m)$ time besides solving some QP subproblem. On the other hand, the size of each QP subproblem is bounded by $O(\delta^2)$. Hence, the overall time bound is $O((\delta \ln m)(m + \mathrm{T}_{\mathrm{QP}}(n, \delta^2))$ $= O(\ln m(\delta m + \delta^4 n^2))$. Here δ is a new parameter called a "combinatorial dimension"; this δ is $n+1$ if no outlier exists, but it could be large if there are many outliers. Therefore, our algorithm has an advantage (over simply solving the original QP problem) if $n \ll m$ and δ can be bounded by $O(n)$. In this sense, our algorithm as it is

may not be practical. But we hope that our approach and analysis of the obtained algorithm could be used for theoretical investigation on the performance of the other similar algorithmic approaches.

Although the above worst case time bound is not appealing, it is still possible that our algorithm performs well even in some practical cases. Unfortunately, our proved upper bound for δ is $\min((n+1)/D, m)$; but we conjecture that it is small (i.e., almost linearly bounded by n) in some cases, in particular, after removing very bad outliers. Also although our theoretical analysis is made on some fixed QP subproblem size (i.e., $O(\delta^2)$), we may choose much smaller subproblems. We can guarantee that the output of our algorithm (if it yields) is correct no matter how δ is set and/or the subproblem size is chosen; the only problem is that the algorithm does not terminate in the expected number of steps. Thus, by running the algorithm from some small δ and some small subproblem size, and by doubling these parameters if the algorithm does not seem to terminate, we can execute the algorithm with quite reasonable parameters. This type of experimental studies are our future work.

2 Support Vector Machines, Optimization, and Random Sampling

Here we explain basic notions on SVM and random sampling techniques. We mainly explain those necessary for our discussion. For SVM, see, e.g., the textbook [16] or the survey [6]; and for random sampling techniques, see the survey [20].

The SVM training for the two-class classification problem can be phrased as follows. Given a set of labeled examples, we have to determine a hyperplane separating positive and negative examples with the largest possible margin, i.e., maximal separation from all the data points.

A possible formulation for the problem is presented as follows. Suppose that we are given a set of m examples $\boldsymbol{x}_i$, $1 \le i \le m$, in an n-dimensional space, say $\mathbb{R}^n$. Each example $\boldsymbol{x}_i$ is labeled by $y_i \in \{1, -1\}$ denoting the classification of the example. We assume here that the data set is linearly separable, i.e., a hyperplane separating the two classes of examples exists; the nonseparable case, which is our main object, will be discussed shortly. The *SVM training problem* that will be discussed in this paper is essentially to solve the following optimization problem (P1).

Max Margin (P1)

$$\begin{array}{lll} \text{min.} & \dfrac{1}{2}\|\boldsymbol{w}\|^2 - (\theta_+ - \theta_-) & \\ \text{w.r.t.} & \boldsymbol{w} = (w_1, \ldots, w_n),\ \theta_+, \text{ and } \theta_-, & \\ \text{s.t.} & \boldsymbol{w} \cdot \boldsymbol{x}_i \ge \theta_+, & \text{if } y_i = 1, \quad \text{and} \\ & \boldsymbol{w} \cdot \boldsymbol{x}_i \le \theta_-, & \text{if } y_i = -1. \end{array}$$

Here we follow [5] and use their formulation. The problem can be restated with a single threshold parameter as given in the original paper [15].

Remarks on Notations Throughout this paper, we use X to denote the set of examples, and let n and m 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 $\{1, \ldots, m\}$.

By the *solution* of (P1), we mean the hyperplane that achieves the minimum cost. We sometimes consider a partial problem of (P1) that minimizes the target cost under some subset of constraints. A solution to such a partial problem of (P1) is called a *local solution* of (P1) for the subset of constraints. Given a solution, its support vectors are the data points $\boldsymbol{x}_i$ for which, at the solution, the corresponding inequality is tight; that is, $\boldsymbol{w} \cdot \boldsymbol{x}_i = \theta_+$ if $y_i = 1$, and $\boldsymbol{w} \cdot \boldsymbol{x}_i = \theta_-$ if $y_i = -1$.

Intuitively, the maximal margin separator does not necessarily lie towards either class, and therefore provides an idea why it could generalize better. Formal discussions can be found in [16], where it is proved that, in the linearly separable case, the generalization error is bounded by a term that depends on the margin but not on the dimensionality of the space. In fact, the dimension could be actually infinite if a way to operate with the corresponding vectors would exist, and the trained SVM would still obey a reasonable bound on the generalization error.

An important feature of SVM is that it is also applicable for the nonseparable case. More precisely, for nonseparable data we can take two positions: (i) the case where we consider that a hyperplane is too weak to be a classifier for our given examples, and that we should be able to fit them better nonlinearly; and (ii) the case where we consider that there are some erroneous examples or exceptions, i.e., "outliers", which should be somehow identified and allowed to be misclassified. Of course, it would be better if we can use a nonlinear classifier. Nevertheless, the second approach is important as well if we suspect that outliers exist in a given set of examples. The usability of SVM is due to the fact that we can use both.

The first subcase is solved by the SVM approach by mapping examples into a much higher dimension space; we come back to this point later on. The second subcase is solved by relaxing constraints by introducing slack variables or "soft margin error". That is, we consider the following generalization of the problem (P1), corresponding to the soft margin hyperplane separation problem.

Max Soft Margin (P2)

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

Again this formulation from [5] is different from the standard one [15]. But it is shown [5] that these two formulations are equivalent provided that D is small enough so that a nontrivial solution exists.

For a given set X of examples, suppose we solve the problem (P2) and obtain the optimal hyperplane. Then an example in X is called an *outlier* if it is misclassified with this hyperplane. On the other hand, examples other than outliers are called *normal examples*. Throughout this paper, we use ℓ to denote the number of outliers.

Notice that this definition of outlier is relative both to the hypothesis class and to the soft margin parameter D, which determines the degree of influence of the outliers. Note that D should be fixed in advance; that is, D is a constant throughout the training process. This point will be discussed later.

Again, the concept of support vector for (P2) is defined in terms of tight inequalities: that is, $\boldsymbol{w} \cdot \boldsymbol{x}_i = \theta_+ - \xi_i$ if $y_i = 1$, and $\boldsymbol{w} \cdot \boldsymbol{x}_i = \theta_- + \xi_i$ if $y_i = -1$.

For the soft margin case, some generalization error bounds exist but tend to be weaker, and/or depend on particular properties of the underlying probability distribution. The formulation, however, turns out to be close to those obtained from applying regularization theory to the problem of fitting the examples as well as possible [17, 18]: the sum of slacks corresponds to minimizing the error, whereas the norm of the vector corresponds to a regularization term, with regularizer factor $1/D$; except that the bias terms do not correspond to such a regularized problem.

2.1 LP-type Optimization Problems and the Sampling Lemma

We explain now, briefly, the essentials of the abstract framework for discussing randomized sampling techniques that was given by Gärtner and Welzl [20]. (The idea, and its algorithmic application as in our Theorem 2.2 below, can be found already in the paper by Clarkson [13], where a randomized algorithm for linear programming has been proposed. Indeed, a similar idea has been used [1] to design an efficient randomized algorithm for quadratic programming. Here we explain the framework and the algorithm following [20].)

Randomized sampling techniques, particularly, the Sampling Lemma below, are 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. In the case of our problem (P1), for example, we can consider an LP-type problem $(\mathcal{D}_1, \phi_1)$, where $\mathcal{D}_1$ is X, and ϕ_1 is a mapping from a given subset X_R of X to the local solution of (P1) for the subset of constraints corresponding to X_R. Of course, there is a certain set of conditions [20] that any LP-type problem must satisfy. But we omit the explanation here and simply mention that our $(\mathcal{D}_1, \phi_1)$ clearly satisfies these conditions.

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. For the problem (P1), each basis is a minimal set of support vectors. Hence, the combinatorial dimension of (P1) is at most $n + 1$, which is due to the fact that the number of support vectors for (P1) is at most $n + 1$

Consider any LP-type problem, and any subset $\mathcal{R}$ of $\mathcal{D}$. 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 in* $\mathcal{R}$ if $\phi(\mathcal{R} - \{e\}) \neq \phi(\mathcal{R})$. In our case, for any subset X_R of X, let $(\boldsymbol{w}, \theta_+, \theta_-)$ be a local solution of (P1) obtained for X_R. Then $\boldsymbol{x}_i \in X$ is a *violator* of X_R (or, more directly, a *violator* of $(\boldsymbol{w}, \theta_+, \theta_-)$) if the constraint corresponding to $\boldsymbol{x}_i$ is not satisfied by $(\boldsymbol{w}, \theta_+, \theta_-)$.

Consider again any LP-type problem $(\mathcal{D}, \phi)$. Let $\mathcal{U}$ be a set consisting of u elements of $\mathcal{D}$. $\mathcal{U}$ may be a *multiple set*, i.e. a set containing possibly some elements

more than once. In order to discuss the case when elements of $\mathcal{D}$ are chosen into $\mathcal{R}$ according to some possibly nonuniform probability, we will use $\mathcal{U}$ as domain instead of $\mathcal{D}$, and will consider simply that $\mathcal{R}$ is a subset of $\mathcal{U}$. Though obvious, the following relation is important for our discussion.

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

Define $v_{\mathcal{R}}$ and $x_{\mathcal{R}}$ to be the number of violators and extremes of $\mathcal{R}$ in $\mathcal{U}$ respectively. The following bound, which is also easy from the definition, is important.

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

We are ready to state the Sampling Lemma. (Here, for the completeness, we present the proof given in [20].)

Lemma 2.1 *Let $(\mathcal{D}, \phi)$ be any LP-type problem. Assume some weight scheme u on $\mathcal{D}$ that gives an integer weight to each element of $\mathcal{D}$. Let $u(\mathcal{D})$ denote the total weight. For a given r, $0 \leq r < u(\mathcal{D})$, we consider the situation where a set of r elements of $\mathcal{D}$ has been chosen randomly, according to their weights. Let $\mathcal{R}$ denote the set of chosen elements, and let $v_{\mathcal{R}}$ be the weight of violators of $\mathcal{R}$. Then we have the following bound on the expected value of $v_{\mathcal{R}}$:*

$$\mathrm{Exp}(v_{\mathcal{R}}) \leq \frac{u(\mathcal{D}) - r}{r+1} \cdot \delta. \tag{3}$$

Proof For a given weight scheme u on $\mathcal{D}$, we define $\mathcal{U}$ as a multiple set containing exactly $u(x)$ copies of each element x in $\mathcal{D}$. Then choosing a set $\mathcal{R}$ of r elements randomly from $\mathcal{D}$ according to their weights is essentially the same as choosing one set $\mathcal{R}$ from all possible $\binom{u(\mathcal{D})}{r}$ (multi)subsets of $\mathcal{U}$ uniformly at random.

For randomly selected $\mathcal{R}$, consider the values $v_{\mathcal{R}}$ and $x_{\mathcal{R}}$. Let v_r and x_r denote their expected values $\mathrm{Exp}(v_{\mathcal{R}})$ and $\mathrm{Exp}(x_{\mathcal{R}})$. Then we prove

$$(r+1)v_r = x_{r+1}(u(\mathcal{D}) - r). \tag{4}$$

The bound of the lemma follows from this and the bound (2).

The above relation (4) is derived easily from the following. Here we use $\binom{\mathcal{U}}{r}$ to denote the set of all r-element subsets of $\mathcal{U}$.

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

□

See [20] for additional explanations, variations for other sampling schema, important related results such as tail bounds, and a large number of incarnations of this Sampling Lemma.

2.2 Preliminary Algorithmics

Consider first the separable case (P1). We can solve this optimization problem by using a standard general quadratic programming algorithm. In some applications, however, the number m of examples is much larger than the dimension n (in other words, many more constraints than variables). This is the situation where randomized sampling techniques are effective.

We first describe how to adapt the general-purpose randomized algorithm from [20], which works for arbitrary LP-type problems. The adaptation of both the algorithm and its analysis is straightforward, but it serves here the purpose of a preliminary easier case that simplifies later on the discussion of our new algorithm.

The idea is simple. Pick up a certain number of examples from X and solve (P1) under the set of constraints corresponding to these examples. We choose examples randomly according to their "weights", where initially all examples are given the same weight. Clearly, the obtained local solution is, in general, not the global solution, and it does not satisfy some constraints; in other words, some examples are misclassified by the local solution. Then double the "weight" of such misclassified examples, and then pick up some examples again randomly according to their weights. If we iterate this process several rounds, the weight of "important examples", which are support vectors in our case, grows exponentially fast, and hence, they are likely to be chosen. Note that once all support vectors are chosen at some round, then the local solution of this round is the true one, and the algorithm terminates at this point. By using the Simple Sampling Lemma, we can prove that the algorithm terminates in $O(n \log m)$ rounds on average.

Now we present in more detail the algorithm in Fig. 1. We use u to denote a weight scheme that assigns some integer weight $u(\boldsymbol{x}_i)$ to each $\boldsymbol{x}_i \in X$. For this weight scheme u, consider a multiple set U containing each example $\boldsymbol{x}_i$ exactly $u(\boldsymbol{x}_i)$ times. Note that U has $u(X)$ $(= \sum_i u(\boldsymbol{x}_i))$ elements. Then by "choose r examples randomly from X according to u", we mean to select a set of examples randomly from all $\binom{u(X)}{r}$ subsets of U with equal probability.

For analyzing the efficiency of this algorithm, we use the Simple Sampling Lemma 2.1. From it, we can prove the following bound. (Again we state the proof for the completeness, though it is immediate from the general argument given in [20].)

procedure OptMargin
  set weight $u(\boldsymbol{x}_i)$ to be 1 for all examples $\boldsymbol{x}_i$ in X;
  $r \leftarrow 6\delta^2$; % $\delta \leq n+1$.
  repeat
    $X_R \leftarrow$ choose r examples from X randomly according to u;
    $(\boldsymbol{w}, \theta_+, \theta_-)$ is a solution of (P1) for X_R;
    $V \leftarrow$ the set of violators in X of the solution;
    if $u(V) \leq u(X)/(3\delta)$ **then** double the weight $u(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i \in V$;
  until $V = \emptyset$;
  return the last solution;
end-procedure.

Fig. 1 A first randomized SVM training algorithm

Theorem 2.2 *The average number of iterations executed in the OptMargin algorithm is bounded by* $6\delta \ln m = O(n \ln m)$. (*Recall that* $|X| = m$ *and* $\delta \leq n + 1$.)

Proof We say a repeat-iteration is *successful* if the if-condition holds in the iteration.

We first bound the number of successful iterations. For this, we analyze how the total weight $u(X)$ increases. Consider the execution of any successful iteration. Since $u(V) \leq u(X)/3\delta$, by doubling the weight of all examples in V, i.e., all violators, $u(X)$ increases by at most $u(X)/(3\delta)$. Since $u(X)$ is initially m, after t successful iterations, we have $u(X) \leq m(1 + 1/(3\delta))^t$.

Let $X_0 \subseteq X$ be a fixed minimal set of support vectors of (P1). X_0 constitutes a basis, and thus it defines the same solution hyperplane as X; thus, if all elements of X_0 are chosen into X_R, i.e., $X_0 \subseteq X_R$, then there is no violator for X_R. At each successful iteration (if it is not the end) some $\boldsymbol{x}_i$ of X_0 must not be in X_R, which implies that it is a violator of X_R by the fact that X_0 is a basis. Hence, $u(\boldsymbol{x}_i)$ gets doubled. Since $|X_0| \leq \delta$, there is some $\boldsymbol{x}_i$ in X_0 that gets doubled at least once every δ successful iterations. Therefore, after t successful iterations, $u(\boldsymbol{x}_i) \geq 2^{t/\delta}$.

Therefore, we have the following upper and lower bounds for $u(X)$.

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

This implies that $t < 3\delta \ln m$ as long as we have not reached the last iteration. That is, the algorithm terminates within less than $3\delta \ln m$ successful iterations.

Next we must estimate how often a successful iteration occurs. Here we use the Sampling Lemma. Consider the execution of any repeat-iteration. Let u be the current weight on X, and let R and V be the set chosen at this iteration and the set of violators of X_R. Then this X_R corresponds to $\mathcal{R}$ in the Sampling Lemma, and we have $u(V) = v_{\mathcal{R}}$. Hence from the above bound 3, we can bound the expectation of $u(V)$ by $(u(X) - r)\delta/(r + 1)$, which is smaller than $u(X)/(6\delta)$ by our choice of r. Thus, the if-condition is satisfied if $u(V)$ is less than double its own average, which happens with probability 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 this randomized OptMargin algorithm needs to solve some subproblem of (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, then this algorithm is faster than solving (P1) directly.

We want to apply a similar technique for the nonseparable case. It seems that we can simply apply the same sampling technique for solving (P2). But this approach is not so straightforward. For example, we need to redefine the notion of "violator". Notice that for any solution hyperplane (even the optimal one), there must be some examples that are misclassified by it. Thus, we cannot simply regard a misclassified example as a violator. Intuitively, it seems reasonable to consider an example as a violator to the current local solution if it is misclassified by the solution *and* it is not used to obtain the solution. It turns out that this intuitive approach works. Our first main technical contribution is to give a formal justification to this intuition and derive a random sampling based algorithm for (P2). We will do so by using alternative

formulations of (P2) given in [5], which also helps us to understand the structure of the optimal solution of (P2).

For designing SVM training algorithms, it is also important that the computation can be done in such a way that the only operations acting on the data points are scalar products, and that the output hyperplane can be defined as a linear combination of the data points. These are necessary for us to use kernels mapping into feature spaces and to be able to use much more complex classifiers [16]. We will see that our derived algorithm satisfies these requirements.

3 Alternative Formulations

To go on we need an alternative formulation of (P2), which is derived based on an intuitive geometric interpretation of (P2) that has been given by Bennett and Bredensteiner [5].

3.1 Derivation of the Formulation

For the completeness, we review the alternative formulations of (P2) given in [5]. First, it is shown [5] that (P2) is equivalent to the following problem (P3). (More precisely, the problem (P3) is the Wolfe dual of (P2).)

Reduced Convex Hull (P3)

$$\text{min.} \quad \frac{1}{2}\left\|\sum_i y_i s_i \boldsymbol{x}_i\right\|^2 \quad \text{w.r.t. } s_1, \ldots, s_m,$$

$$\text{s.t.} \quad \sum_{i:\, y_i=1} s_i = 1, \quad \sum_{i:\, y_i=-1} s_i = 1, \quad \text{and} \quad 0 \le s_i \le D.$$

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 hulls of positive and negative examples. In the separable case, it is the distance between two closest points. On the other hand, in the nonseparable case, we give some restriction to the influence of each example and consider only points defined by examples in such a way that each example cannot contribute more than D.

As mentioned in [5], 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 paper, we use k to denote this constant.) Then our objective can be regarded as the distance between the convex hulls of "composed examples", points that are defined as a center of k examples. Then resulting convex hulls are reduced ones and they may be separable by some hyperplane. In the extreme case where $k = m_+ = m_-$ (where m_+ and m_- are respectively the number s of positive and negative examples), we have only one

positive and one negative composed examples, which are clearly separable (unless they are the same).

Remarks on the Choise of D Although we will assume that $D = 1/k$ for some integer $k > 0$, this is simply for our explanation and this restriction is not essential for executing our algorithm. Also note that we do not have to use a single constant for the soft margin parameter D. In particular, it would be more reasonable to use two D_+ and D_- (hence, k_+ and k_-) for positive and negative examples. But for simplifying our notation, we will discuss with a single soft margin parameter D throughtout the paper. The modification of our algorithm to a two or more soft margin parameter cases is easy.

We state the above intuition formally by reformulating (P3). Let 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 $\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \ldots, \boldsymbol{x}_{i_k}$ of X with the same label (i.e., $y_{i_1} = y_{i_2} = \cdots = y_{i_k}$). That is, each composed example is the mass center of a group of k homogeneously labeled initial data points. The label y_I of the composed example $\boldsymbol{z}_I$ inherits its members'. Throughout this note, we use I for indexing elements of Z and their labels. The range of I is $\{1, \ldots, M\}$, where $M \stackrel{\text{def}}{=} |Z|$. 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. (For distinguishing from composed examples $\boldsymbol{z}_I$, we will call $\boldsymbol{x}_i$ an *original example*.)

Then it is easy to see that (P3) is equivalent to the following (P4).

Convex Hull of Reduced Points (P4)

$$\text{min.} \quad \frac{1}{2}\left\| \sum_I y_I s_I \boldsymbol{z}_I \right\|^2 \quad \text{w.r.t. } s_1, \ldots, s_M,$$

$$\text{s.t.} \quad \sum_{I:\, y_I = 1} s_I = 1, \quad \sum_{I:\, y_I = -1} s_I = 1, \quad \text{and} \quad 0 \leq s_I \leq 1.$$

More formally, the same optimal value is achieved by solutions $\{s_i^*\}_{i=1,\ldots,m}$ and $\{s_I^*\}_{I=1,\ldots,M}$ satisfying the following for all i, $1 \leq i \leq m$.

$$\frac{1}{k}\left(\sum_{I:\boldsymbol{x}_i \in z_I} s_I^* \right) = s_i^*. \tag{5}$$

This is because for any solution $\{s_i^*\}_{i=1,\ldots,m}$, we can find $\{s_I^*\}_{I=1,\ldots,M}$ satifying the above for all i.

In this paper, we further consider the Wolfe primal of this problem again. Then we come back to the one almost identical to (P1)!

Max Margin for Composed Examples (P5)

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

In general, composed examples may not be separable. But they are separable for sufficiently large k, in which case it can be shown that (P2) and (P5) are essentially equivalent. More precisely, from the observation given in [5], we can show the following relationship between (P2) and (P5).

Theorem 3.1 *Suppose that k is large enough so that the obtained composed examples are linearly separable. Then the weight vector for the optimal separating hyperplane, i.e., $\boldsymbol{w}^*$, coincides for* (P2) *and* (P5). (*Note that the margin parameters (θ_+^*, θ_-^*) and (η_+^*, η_-^*) are usually different.*)

Note that the problem (P5) is essentially the same as (P1); thus, we can use our randomized algorithm OptMargin of Fig. 1 for (P5). In fact, the problem (P5) is regarded as the LP-type problem $(\mathcal{D}_5, \phi_5)$ similar to $(\mathcal{D}_1, \phi_1)$ for (P1), where $\mathcal{D}_5$ is Z, and ϕ_5 is a mapping from a given subset Z_R of Z to the local solution of (P5). In particular, the combinatorial dimension of (P5) is $n + 1$, the same as that of (P1). Only the difference is that there are now $M = O(m^k)$ composed examples (hence, so many constraints). This number is quite large, but this is the situation suitable for the sampling technique.

Suppose now that we use OptMargin for solving (P5). From our analysis, the expected number of iterations is $O(n \ln M) = O(kn \ln m)$. That is, we need to solve QP problems with $n + 2$ variables and $O(n^2)$ constraints for $O(kn \ln m)$ times. Although this is not bad at all, there are unfortunately some serious problems. The algorithm needs, at least as it is, a large amount of time and space for "book keeping" computation. First of all, we have to keep weights of all M composed examples in Z. Secondly, for finding violators and for modifying weights, we have to go through Z, which takes at least $O(M)$ steps. Also it is not so easy to choose composed examples randomly according to their weights. Some solution to these problems were proposed in [2], but the proof of the running time of the resulting algorithm depended on an unproven hypothesis.

In this paper, instead of using OptMargin directly to (P5), we derive an algorithm, based on a nontrivial geometric lemma, that handles only m weights and avoids searching for violators on all of Z. In fact, this algorithm is a natural generalization of OptMargin for (P2). Also since it uses only scalar products on data points, it combines with any desired kernel.

3.2 Properties of the Solutions

Before deriving the algorithm, let us first examine solutions to (P2) and (P5). Throughout this subsection, we assume that k is large enough so that the set of composed examples are linearly separable.

For a given example set X, let Z be the set of composed examples. Let $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \xi^*)$ and $(\boldsymbol{w}^*, \eta_+^*, \eta_-^*)$ be the solutions of (P2) for X and (P5) for Z respectively, sharing $\boldsymbol{w}^*$ as explained above. Let $X_{\mathrm{err},+}$ and $X_{\mathrm{err},-}$ denote the sets of positive/negative outliers. That is, $\boldsymbol{x}_i$ belongs to $X_{\mathrm{err},+}$ (resp., $X_{\mathrm{err},-}$) 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 ℓ_+ and ℓ_- to denote the number of positive/negative outliers.

Under the LP-type interpretation $(\mathcal{D}_5, \phi_5)$ of (P5), consider any fixed basis Z_0 of Z. In order to facilitate understanding, we assume nondegeneracy throughout the following discussion. Note that every element of the basis is extreme in Z. Hence, we call elements of Z_0 *final extremers*. By definition, the solution of (P5) for 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 prove the following lemma. (For the sake of simplicity, this lemma is stated only for the positive examples; but the corresponding properties clearly hold for the negative examples.)

Lemma 3.2 *We use symbols defined above. In particular, assume that composed examples in Z are linearly separable under the current choice of k. Let $\boldsymbol{z}_I$ be any positive final extremer, i.e., an element of Z_0 such that $y_I = 1$. Then the following properties hold.*

(a) $\boldsymbol{w}^* \cdot \boldsymbol{z}_I = \eta_+^*$.
(b) $X_{\mathrm{err},+} \subseteq z_I$. *Hence*, $\ell_+ \leq k$.
(c) *We may consider that* $X_{\mathrm{err},+} \neq z_I$, *i.e.*, $\ell_+ < k$.
(d) *For every* $\boldsymbol{x}_i \in z_I$, *if* $\boldsymbol{x}_i \notin X_{\mathrm{err},+}$, *then we have* $\boldsymbol{w}^* \cdot \boldsymbol{x}_i = \theta_+^*$.
(e) *If* $X_{\mathrm{err},+}$ *is not empty, then we have* $\eta_+^* < \theta_+^*$.

Proof (a) Since Z_0 is the set of final extremers, (P5) can be solved only with the constraints corresponding to elements in Z_0. Suppose that $\boldsymbol{w}^* \cdot \boldsymbol{z}_J > \eta_+^*$ for some positive $\boldsymbol{z}_J \in Z_0$ including $\boldsymbol{z}_I$ of the lemma. Let Z' be the set of such $\boldsymbol{z}_J$'s of Z_0. If Z' indeed contained all positive examples in Z_0, then we could set η_+ with $\eta_+^* - \epsilon$ for some $\epsilon > 0$ and still satisfy all the constraints, which contradicts the optimality of the solution. Hence, we may assume that $Z_0 - Z'$ still has some positive example. Then it is well known (see, e.g., [8]) that a local optimal solution to the problem (P5) with the constraints corresponding to elements in Z_0 is also locally optimal to the problem (P5) with the constraints corresponding to only elements in $Z_0 - Z'$. Furthermore, since (P5) is a convex programming, a local optimal solution is globally optimal. Thus, the original problem (P5) is solved with the constraints corresponding to elements in $Z_0 - Z'$. This contradicts our assumption that Z_0 is the set of final extremers.

(b) Since (P2) is a convex minimization problem, the KKT-point of (P2) is obtained from its solution $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \xi^*)$. That is, with some $(\boldsymbol{s}^*, \boldsymbol{u}^*)$, the tuple

$(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \xi^*, \boldsymbol{s}^*, \boldsymbol{u}^*)$ satisfies the following so called KKT-condition. (Below we use i to denote indices of examples, and let P and N respectively denote the set of 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.)

$$\boldsymbol{w}^* - \sum_{i\in P} s_i \boldsymbol{x}_i + \sum_{i\in N} s_i^* \boldsymbol{x}_i = 0, \qquad D\boldsymbol{e} - \boldsymbol{s}^* - \boldsymbol{u}^* = 0,$$

$$-1 + \sum_{i\in P} s_i^* = 0, \qquad -1 + \sum_{i\in N} s_i^* = 0,$$

$$\forall i \in P\ [s_i^*(\boldsymbol{w}^* \cdot \boldsymbol{x}_i - \theta_+^* + \xi_i^*) = 0], \qquad \forall i \in N\ [s_i^*(\boldsymbol{w}^* \cdot \boldsymbol{x}_i - \theta_-^* - \xi_i^*) = 0],$$

$$\boldsymbol{u}^* \cdot \xi^* = 0 \text{ (which means } (D\boldsymbol{e} - \boldsymbol{s}^*) \cdot \xi^* = 0), \text{ and } \xi^*, \boldsymbol{u}^*, \boldsymbol{s}^* \geq 0.$$

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 $\boldsymbol{u}^* \geq 0$.)

$$\boldsymbol{w}^* = \sum_{i\in P} s_i^* \boldsymbol{x}_i - \sum_{i\in N} s_i^* \boldsymbol{x}_i,$$

$$\sum_{i\in P} s_i^* = 1, \quad \sum_{i\in N} s_i^* = 1, \quad \text{and} \quad 0 \leq \boldsymbol{s}^* \leq D\boldsymbol{e}.$$

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

Consider any $\boldsymbol{x}_i \in X_{\mathrm{err},+}$; we will show below that $\boldsymbol{x}_i \in z_I$ for any $\boldsymbol{z}_I \in Z_0$. Since $\xi_i^* > 0$, from the requirements that $(D\boldsymbol{e} - \boldsymbol{s}^*) \cdot \xi^* = 0$ and that $D\boldsymbol{e} - \boldsymbol{s}^* \geq 0$ it follows that $D - s_i^* = 0$; that is, $s_i^* = 1/k$. Then from (5) we have $\sum_{I:\boldsymbol{x}_i \in z_I} s_I^* = 1$. On the other hand, from the constraint of (P4), we have $\sum_{I:y_I=1} s_I^* = 1$. Hence, $s_I^* > 0$ implies $\boldsymbol{x}_i \in z_I$ for any I. Now by the equivalence of (P4) and (P5), we see that the final extremers are exactly points contributing to the solution of (P4). That is, for any I, it holds that $\boldsymbol{z}_I \in Z_0$ if and only if $s_I^* > 0$. Therefore, for any $\boldsymbol{z}_I \in Z_0$, we have $\boldsymbol{x}_i \in z_I$.

(c) If Z_0 contains more than one positive element, then the relation $X_{\mathrm{err},+} \neq z_I$ is immediate from the above. A subtle case is when $\boldsymbol{z}_I$ is the unique positive element of Z_0. Suppose that all $\boldsymbol{x}_i \in z_I$ are outliers w.r.t. $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \xi^*)$. In this case, however, the same optimal value is obtained by using $\theta_+ = \boldsymbol{w}^* \cdot \boldsymbol{x}_i$ as θ_+^* for any $\boldsymbol{x}_i \in z_I$. Thus, we may assume that θ_+^* is the smallest $\boldsymbol{w}^* \cdot \boldsymbol{x}_i$, in which case there is at least one element not in $X_{\mathrm{err},+}$.

(d) Consider any index i in P such that $\boldsymbol{x}_i$ appears in some of the final extremer $\boldsymbol{z}_I \in Z_0$. Since $s_I^* > 0$, we can show that $s_i^* > 0$ by using the equation (5). 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 $\boldsymbol{x}_i \notin X_{\mathrm{err}}$, i.e., it is not an outlier or $\xi_i^* = 0$, then we have $\boldsymbol{w}^* \cdot \boldsymbol{x}_i = \theta_+^*$.

(e) From (a), we have

$$\eta_+^* = \boldsymbol{w}^* \cdot \frac{\boldsymbol{x}_{i_1} + \boldsymbol{x}_{i_2} + \cdots + \boldsymbol{x}_{i_k}}{k}.$$

Since $X_{\mathrm{err},+}$ is not empty, some of these $\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_k}$ are outliers; for such an outlier $\boldsymbol{x}_{i_j}$, it holds that $\boldsymbol{w}^* \cdot \boldsymbol{x}_{i_j} < \theta_+^*$. On the other hand, as shown in (c), we have $\boldsymbol{w}^* \cdot \boldsymbol{x}_{i_j} = \theta_+^*$ for all normal examples in Z_0. Thus,

$$\eta_+^* = \boldsymbol{w}^* \cdot \frac{\boldsymbol{x}_{i_1} + \boldsymbol{x}_{i_2} + \cdots + \boldsymbol{x}_{i_k}}{k} < \frac{k\theta_+^*}{k} = \theta_+^*. \qquad \square$$

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_+^*. Essentially, (b) means that, when the final extremers are obtained, the set of outliers can be obtained as $\bigcap_{z_I \in Z_0} z_I$, the set of examples appearing in all final extremers. On the other hand, the fact (d) states that all original normal examples, i.e., examples not in $X_{\mathrm{err},+}$, appearing in some final extremer are all located again on some hyperplane whose distance from the base hyperplane is $\theta_+^* > \eta_+^*$. (Here we assume that $k > \ell_+$ and each final extremer contains $k - \ell_+$ normal examples.) Now consider the point

$$\boldsymbol{v}_+ \stackrel{\mathrm{def}}{=} \frac{1}{\ell_+} \cdot \left(\sum_{\boldsymbol{x}_i \in X_{\mathrm{err},+}} \boldsymbol{x}_i \right),$$

namely, the center of positive outliers. 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 ℓ_+ positive outliers and $k - \ell_+$ normal examples, we have

$$\theta_+^* - \eta_+^* : \eta_+^* - \mu_+^* = \ell_+ : k - \ell_+.$$

4 Our Algorithm

Now we are ready to derive the algorithm for (P2). First we give a rather rough outline of the algorithm, and then discuss its details formally. Throughout this section, we again assume that k $(= 1/D)$ is large enough so that the set of composed examples is linearly separable.

We use the same notations as in the previous sections. X is the set of original examples $\boldsymbol{x}_i$, i.e., input data, and Z is the set of composed examples $\boldsymbol{z}_I$ made up from X. Our goal is to find a maximal margin separator of these composed examples. As explained in the previous section, the application of the algorithm OptMargin to (P5) requires some heavy task such as book keeping of all weights of composed examples. This problem may be solved by associating weights to original examples $\boldsymbol{x}_i$ rather than to composed examples. Instead of generating composed examples according to their weights, we first generate original examples according to their weights, and define the set of composed examples that are made up from the generated original examples. More specifically, we define P_k to be a mapping from any subset X' of X to the set of all composed examples consisting only of k elements of X'. Then

procedure OptMargin_Composed
  set weight $u(\boldsymbol{x}_i)$ to be 1 for all examples $\boldsymbol{x}_i$ in X;
  $r \leftarrow 6\delta^2$;
  loop
    $X_R \leftarrow$ choose r elements from X randomly according to their weights;
    $Z_R \leftarrow P_k(X_R)$;
    $(\boldsymbol{w}, \eta_+, \eta_-) \leftarrow$ the solution of (P5) for Z_R;
    if no composed example is misclassified by $(\boldsymbol{w}, \eta_+, \eta_-)$ **then exit**;
    $V \leftarrow$ the set of "violator" under the interpretation of $(\mathcal{D}_2, \phi_2)$;
    double the weight of examples in V (unless $u(V)$ is too large);
  end loop;
  compute $(\boldsymbol{w}, \theta_+, \theta_-)$ from the last solution $(\boldsymbol{w}, \eta_+, \eta_-)$;
  return $(\boldsymbol{w}, \theta_+, \theta_-)$;
end procedure.

Fig. 2 A new randomized SVM training algorithm (outline)

we consider the following procedure: (1) generate a set X_R of examples from X, (2) compute the set $Z_R = P_k(X_R)$, (3) solve (P5) on Z_R, and (4) if the obtained local solution is not satisfiable, then increase the weight of examples in X that cause a violator of the current solution. For the implementation of this idea, we introduce a new LP-type problem $(\mathcal{D}_2, \phi_2)$, where $\mathcal{D}_2$ is X, and ϕ_2 is a mapping from any subset X_R of X to the solution of (P5) on $P_k(X_R)$. Based on this new interpretation, we can outline our new sampling algorithm as presented in Fig. 2.

There are several points that have to be clarified. First, we need to show that the defined $(\mathcal{D}_2, \phi_2)$ is indeed an LP-type problem. Intuitively, $(\mathcal{D}_2, \phi_2)$ is parallel to the LP-type problem $(\mathcal{D}_5, \phi_5)$ defined for (P5). In fact, $\phi_2(X_R) = \phi_5(P_k(X_R))$, and $(\mathcal{D}_2, \phi_2)$ can be considered as a special case of $(\mathcal{D}_5, \phi_5)$, where we focus on the sets defined by $P_k(X_R)$ with $X_R \subseteq X$ $(= \mathcal{D}_2)$ for a subset of Z $(= \mathcal{D}_5)$. In order to show that $(\mathcal{D}_2, \phi_2)$ is indeed parallel to $(\mathcal{D}_5, \phi_5)$ and hence it satisfies the conditions for LP-type problems, it is important to establish the following relationship: for any X_R, $Z_R = P_k(X_R)$ has a violator in the sense of ϕ_5 if and only if X_R has a violator in the sense of ϕ_2. This relation is proved by using Lemma 3.2.

Lemma 4.1 *For any $X_R \subseteq X$, let $Z_R = P_k(X_R)$. Also let $(\boldsymbol{w}, \theta_+, \theta_-)$ and $(\boldsymbol{w}, \eta_+, \eta_-)$ be respectively the solution of* (P2) *on X_R and* (P5) *on Z_R. Then the following three statements are equivalent.*

(i) *There exists a composed example $\boldsymbol{z}_I \in Z$ that is misclassified by $(\boldsymbol{w}, \eta_+, \eta_-)$. (Thus, this $\boldsymbol{z}_I$ is a violator of Z_R under the interpretation of $(\mathcal{D}_5, \phi_5)$.)*
(ii) *There exists an original example $\boldsymbol{x}_i \in X - X_R$ such that $P_k(X_R \cup \{\boldsymbol{x}_i\})$ contains a composed example that is misclassified by $(\boldsymbol{w}, \eta_+, \eta_-)$. (Thus, this $\boldsymbol{x}_i$ is a violator of X_R under the interpretation of $(\mathcal{D}_2, \phi_2)$.)*
(iii) *There exists an original example $\boldsymbol{x}_i \in X - X_R$ that is misclassified by $(\boldsymbol{w}, \theta_+, \theta_-)$.*

Remark Precisely speaking, for the solution $(\boldsymbol{w}, \theta_+, \theta_-)$, we consider the one defined in the proof of Lemma 3.2(c).

Proof Since (ii) $\Rightarrow$ (i) is trivial, we show that (i) $\Rightarrow$ (iii) and (iii) $\Rightarrow$ (ii). Here we consider only positive examples, the negative case being analogous.

In the following proof, we will make use of Lemma 3.2. Notice that Z_R is the set of *all* composed examples made from X_R. Thus, Lemma 3.2 holds with $X = X_R$ and $Z = Z_R$.

(i) $\Rightarrow$ (iii): Suppose that there exists a positive composed example $\boldsymbol{z}_I$ that is misclassified by $(\boldsymbol{w}, \eta_+, \eta_-)$; that is, $\boldsymbol{w} \cdot \boldsymbol{z}_I < \eta_+$. Pick any final extremer $\boldsymbol{z}_0$ of Z_R. Then we argue first that $\boldsymbol{z}_I$ contains some misclassified example $\boldsymbol{x}_i$, i.e., an example with $\boldsymbol{w} \cdot \boldsymbol{x}_i < \theta_+$, that is not in $\boldsymbol{z}_0$. Suppose otherwise; that is, all misclassified original examples of $\boldsymbol{z}_I$ are accounted for in $\boldsymbol{z}_0$. Since some correctly classified example $\boldsymbol{x}_j$ exists in $\boldsymbol{z}_0$, and it satisfies $\boldsymbol{w} \cdot \boldsymbol{x}_j = \theta_+$ (Lemma 3.2(c), (d)), this would imply $\boldsymbol{w} \cdot \boldsymbol{z}_I \geq \boldsymbol{w} \cdot \boldsymbol{z}_0$. But this contradicts the assumption $\boldsymbol{w} \cdot \boldsymbol{z}_I < \eta_+$, because $\boldsymbol{w} \cdot \boldsymbol{z}_0 = \eta_+$.

(iii) $\Rightarrow$ (ii): Suppose that there exists a positive example $\boldsymbol{x}_i \in X - X_R$ that is misclassified by $(\boldsymbol{w}, \theta_+, \theta_-)$. This means that $\boldsymbol{w} \cdot \boldsymbol{x}_i < \theta_+$. Consider any final extremer $\boldsymbol{z}_0$ of Z_R. Then its corresponding inequality is tight; that is, $\boldsymbol{w} \cdot \boldsymbol{z}_0 = \eta_+$. Also it follows from Lemma 3.2(b) and (c) that $\boldsymbol{z}_0$ contains all misclassified original examples of X_R and that the remaining examples $\boldsymbol{x}_j \in \boldsymbol{z}_0$ fulfill $\boldsymbol{w} \cdot \boldsymbol{x}_j = \theta_+$. Construct $\boldsymbol{z}_I$ by replacing one of such $\boldsymbol{x}_j$'s with $\boldsymbol{x}_i$. Then we have $\boldsymbol{w} \cdot \boldsymbol{z}_I < \boldsymbol{w} \cdot \boldsymbol{z}_0 = \eta_+$. Hence $\boldsymbol{z}_I$ is misclassified by $(\boldsymbol{w}, \eta_+, \eta_-)$; furthermore, $\boldsymbol{z}_I$ is in $P_k(X_R \cup \{\boldsymbol{x}_i\})$. □

This lemma provides us with a way to find violators of the generated set X_R in our algorithm. That is, if and only if it is not in X_R *and* it is misclassified by the solution of (P2) on X_R. Thus, by solving (P2) on X_R and by using its solution, we can compute the set V violators of X_R. Furthermore, the lemma also guarantees that no composed example is misclassified (by the current solution for Z_R) if and only if $V = \emptyset$. Hence, the stopping condition can be checked by testing whether $V = \emptyset$. This means that we do not have to solve (P5) on Z_R; in fact, it is even not necessary to compute Z_R! These observations lead to our algorithm stated in Fig. 3. (We use the same condition as before for determining when weights should get increased.)

procedure OptSoftMargin
  set weight $u(\boldsymbol{x}_i)$ to be 1 for all examples $\boldsymbol{x}_i$ in X;
  $r \leftarrow 6\delta^2$; % See Lemma 4.2 for the bound for δ.
  repeat
    $X_R \leftarrow$ choose r examples from X randomly according to u;
    $(\boldsymbol{w}, \theta_+, \theta_-)$ is a solution of (P2) for X_R;
    $V \leftarrow$ the set of examples in $X - X_R$ that are misclassified by $(\boldsymbol{w}, \theta_+, \theta_-)$;
    if $u(V) \leq u(X)/(3\delta)$ **then** double the weight $u(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i \in V$;
  until $V = \emptyset$;
  return the last solution;
end-procedure.

Fig. 3 The final randomized SVM training algorithm

Finally, we estimate the combinatorial dimension δ of $(\mathcal{D}_2, \phi_2)$. (See a remark at the end of this section for the tightness of this bound.)

Lemma 4.2 *The combinatorial dimension δ of $(\mathcal{D}_2, \phi_2)$ is at most $k(n+1)$.*

Proof Recall that the combinatorial dimension of $(\mathcal{D}_1, \phi_1)$ is at most $n+1$; then, by the same argument, we can bound the combinatorial dimension of $(\mathcal{D}_5, \phi_5)$ by $n+1$. That is, every basis of $\mathcal{D}_5$ consists of at most $n+1$ composed examples.

Now assume to the contrary that some basis X_1 of $(\mathcal{D}_2, \phi_2)$ has more than $k(n+1)$ elements. Let $Z_1 = P_k(X_1)$. We may assume that both X_1 and Z_1 contain examples from both positive and negative ones so that the problems (P2) and (P5) are nontrivial.[2] Consider any basis Z_2 of Z_1. Then by definition, we have $\phi_2(Z_2) = \phi_2(Z_1)$; in other words, the partial solution of (P2) for Z_2 is the same as the one for Z_1.

Let X_2 be a subset of X_1 such that $Z_2 = P_k(X_2)$. As explained above, since Z_2 has at most $n+1$ composed examples, X_2 has at most $k(n+1)$ examples. Hence, there exists some element $\boldsymbol{x}_0$ in $X_1 - X_2$, which must be a violator of X_2, since X_1 is a basis. In other words, $\boldsymbol{x}_0$ is misclassified by the partial solution of (P2) on X_2. But then from Lemma 4.1, there must be some violator of Z_2 in Z_1, which contradicts the fact $\phi_2(Z_2) = \phi_2(Z_1)$. □

Now we conclude our analysis and estimate the total running time of the algorithm. This can be made in exactly the same way as the one for the separable case (Theorem 2.2), and we have the following theorem.

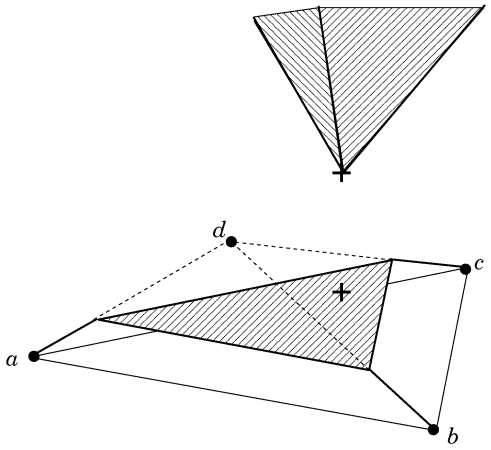
Theorem 4.3 *The average number of iterations executed in the OptSoftMargin algorithm is bounded by $6\delta \ln m = O(n \ln m)$, where $|X| = m$ and $\delta \le k(n+1)$.*

Notice here that our obtained algorithm uses solutions of (P2) for selected original examples. Thus, any algorithm solving (P2) should work; in particular, we can use any kernel method so long as it works for solving the original soft margin problem (P2). But note also that the combinatorial dimension δ may get larger depending on the feature space.

Some Remark on the Combinatorial Dimension The combinatorial dimension δ is an important complexity parameter in our algorithm. First as stated above, it is used to bound the average number of iterations. Also recall that the number r of samples at each iteration—the sampling parameter—is determined by $r = 6\delta^2$. (Though this choice of r is sufficient for proving our bound, our preliminary experiments show that much smaller value of r is sufficient. It is our future work to study how to select the sampling parameter r, in particular, for popular kernels.)

In the above lemma, we give only a simple bound for the combinatorial dimension δ; that is, $\delta \le k(n+1)$. Our experiments, however, indicate that actual δ seems to be much smaller, and it is quite likely that we can give much better bounds for δ. In

[2] We can define our LP-type problems, e.g., $(\mathcal{D}_2, \phi_2)$ so that X_1 has no extreme element (hence, X_1 is not a basis) if its corresponding subproblem of (P2) is trivial.



An example of reduced convex hulls and separating hyperplanes for $n = 3$ and $k = 3$. We assume that all *composed* negative examples are on the surface of the cone (top), while all *composed* positive examples (except three explained below) are under the shadowed triangular plane (bottom). Here we focus on three composed positive examples that are located at vertices of the shadowed triangle. We assume that two *original* positive examples are located at a, b, and c respectively, and that one *original* example exists at d. Then the three composed examples are those cosisitng of two points at a (resp., b and c) and one point at d. The soft margin or the distance between reduced convex hulls is the distance of two points indicated as $+$.

Fig. 4 An example for indicating δ cannot be $n + 1 + k$ (part 1)

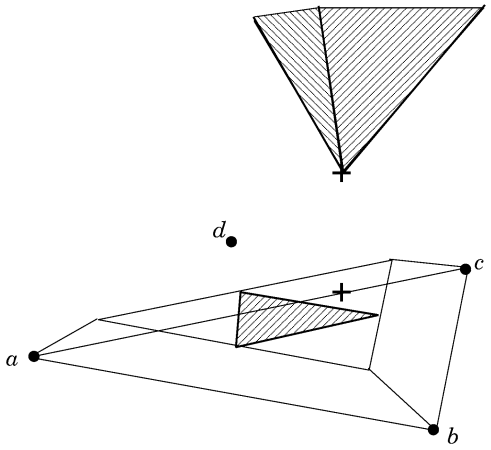
fact, one might argue as follows to bound $\delta \leq n+1+k$: Suppose that for a given set X of examples, the number of outliers to each local solution of (P2) is at most ℓ, where we may assume[3] that $\ell \leq k$ (Lemma 3.2). Then there are at most $n+1+\ell \leq n+1+k$ examples for which the constraint inequality of (P2) is tight. (At most ℓ examples $\boldsymbol{x}_i$ with $\xi_i > 0$, and at most $n+1$ examples $\boldsymbol{x}_i$ with $\xi_i = 0$.) In other words, the solution hyperplane is determined by these at most $n + 1 + k$ examples; hence, it seems that δ is bounded by $n + 1 + k$. Unfortunately, however, an example given in Figs. 4 and 5 suggests that this easy argument would not be correct. (The idea of this example has been pointed out by Léonard Rodriguez.)

Nevertheless, the example is somewhat unconclusive, and we still think that one could show a much better bound for δ. We also leave this problem for our future research topics.

5 Extension to Support Vector Regression

The regression problem under ϵ-insensitive loss by means of SVM can be phrased as follows. We are given a set of points in a product space $X \times Y$, where the coordinates

[3]What we have from Lemma 3.2 is $\ell_+, \ell_- < k$, which guarantees only $\ell < 2k$. But by more precise argument considering positive and negative examples separately, we can indeed show that $\ell \leq k$; the detail is left to the reader.



Suppose that the combinatorial dimension of $(\mathcal{D}_2, \phi_2)$ for the examples explained in Fig. 4 is $n + 1 + k = 7$. Then the same soft margin must be derived by considering some composed examples consisting of only 7 points. Clearly 3 $(= k)$ original examples are needed for the negative side; hence, we can choose at most 4 original examples from the positive side. The shadowed triangle in the figure shows the reduced convex hull when we choose one example from each of a, b, c, and d points. As shown in the figure the distance between two reduced convex hulls is longer than the distance between two + points. The same problem occurs for any choice of 4 original positive examples.

Fig. 5 An example for indicating δ cannot be $n + 1 + k$ (part 2)

in X act as predictor (or given) information and the coordinate in Y act as response (or desired) variable. Also given the value of ϵ for the loss function. We have to determine a hyperplane approximating the cloud of points, in the sense that the hyperplane passes near all the points, missing them all by no more than ϵ on the Y coordinate; thus all are within a so-called ϵ-tube around the hyperplane. Of course, if ϵ is too small or there are some exceptions, this might be impossible, in the same way as classifying with a hyperplane a linearly inseparable dataset. The solution for this case is analogous to the classification approach; that is, soft margins become here slack quantities that allow, but penalize, points that are too far from the hyperplane.

There are several formalizations of the SVM regression problem; here we consider the one studied in [9], and explain briefly the reduction of regression to classification suggested in [9]. (See the references in [9] for the other formulations.)

The formulation we consider here is based on shifting the cloud of points along the coordinate corresponding to the response variable. Indeed, from the data points and from the value ϵ received for the loss function, we construct two new datasets, one by shifting the response up by ϵ, and the other by shifting it down by ϵ. For a large enough ϵ for which hard tubes exist, these shifts amount to move the first dataset to stand fully above the regression hyperplane, and the second dataset to stand fully below it. Thus, the regression hyperplane has been effectively transformed into a classification hyperplane, as was desired. To apply the support vectors approach, we

simply compute the maximal margin separating hyperplane between both translated sets.

Now let us give a precise formulation of the above idea. As before, suppose that we are given a set of m examples $\boldsymbol{x}_i$, $1 \le i \le m$, in some n dimensional space $\mathbb{R}^n$. Each example $\boldsymbol{x}_i$ is associated to a value $y_i \in \mathbb{R}$ of the response variable. Also, the ϵ value for the loss function is given. Based on the above idea, we formulate the (hard tube) *SV regression problem* (SVR) as the optimization problem HardTubeSVR (P6). Note that we are assuming here that ϵ is large enough for the problem to have a solution.

HardTubeSVR (P6)

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

In the same way that the SVM classifiers are also applicable to the nonseparable case, an important feature of SVR is that it can be applied also when the value of ϵ is too small or there exist some exceptions or erroneous examples. This is done in classification by relaxing constraints by introducing slack variables or "soft margin error". The reduction of regression to classification provides a similar solution, and we formulate the *soft tube SV regression problem* as the following SoftTubeSVR (P7). (These problems (P6) and (P7) are respectively problems (5) and (7) in [9].)

SoftTubeSVR (P7)

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

By the *solution* of the problems HardTubeSVR and SoftTubeSVR, we mean the hyperplane that achieves the minimum cost. Given a solution of HardTubeSVR, we have some examples $\boldsymbol{x}_i$—*support vectors*—for which, at the solution, at least one of the corresponding inequalities is tight: $\boldsymbol{w} \cdot \boldsymbol{x}_i + \sigma(y_i + \epsilon) = \theta_+$ or $\boldsymbol{w} \cdot \boldsymbol{x}_i + \sigma(y_i - \epsilon) = \theta_-$. On the other hand, for a solution of SoftTubeSVR, there may exist some examples $\boldsymbol{x}_i$—*outliers*—whose response variables are off the value predicted by the hyperplane by more than ϵ. The softness parameter D determines the degree of influence of the outliers as in the formulation of (P2).

It is easy to transform these optimization problems to those corresponding to the classification problems that we have studied in the previous sections. Then we can simply apply the algorithmics we have developed, and the analysis of the obtained

solution can be made similarly. We explain below the transformation for the hard tube SVR and the soft tube SVR.

Hard Tubes

A slight adjustment of the HardTubeSVR (P6) gives the following equivalent specific classification problem (P1*), where we can see that our examples have both their predictor and response coordinates and that they are considered as positive examples in the $+\epsilon$ case and as negative examples in the $-\epsilon$ case.

$$\begin{array}{ll} \underline{\text{Max Margin (P1*)}} & \\ \text{min.} & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2}\sigma^2 - (\theta_+ - \theta_-) \\ \text{w.r.t.} & \boldsymbol{w} = (w_1, \ldots, w_n),\ \sigma,\ \theta_+,\ \text{and } \theta_-, \\ \text{s.t.} & (\boldsymbol{w}, \sigma) \cdot (\boldsymbol{x}_i, y_i + \epsilon) \geq \theta_+, \quad \text{and} \\ & (\boldsymbol{w}, \sigma) \cdot (\boldsymbol{x}_i, y_i - \epsilon) \leq \theta_-. \end{array}$$

Notice that this formulation is essentially the same as (P1) with $n+1$ dimensionality. Thus, we can solve this (P1*) by using the algorithm OptMargin in Fig. 1, within the running time given by the following theorem.

Theorem 5.1 *The average number of iterations executed in the OptMargin algorithm for* (P1*) *is bounded by* $6(n+2)\ln m = O(n \ln m)$.

As shown in [9], a solution $(\boldsymbol{w}^*, \sigma^*, \theta_+^*, \theta_-^*)$ for (P1*) gives not only the separating hyperplane but also the two biases that indicate both borders (up and down) of the solution hard tube. Bi and Bennett proved (Theorem 3.1 [9]) that the optimal value of ϵ is obtained by $\epsilon^* = \epsilon - (\theta_+^* - \theta_-^*)/(2\sigma^*)$; that is, ϵ^* defines the borders of the hard tube fitting all examples.

Soft Tubes

We can transform, in the same way as above, the SoftTubeSVR (P7) into the following classification problem (P2*).

$$\begin{array}{ll} \underline{\text{Max Soft Margin (P2*)}} & \\ \text{min.} & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2}\sigma^2 - (\theta_+ - \theta_-) + D \cdot \left(\sum_i \xi_i^+ + \sum_i \xi_i^- \right), \\ \text{w.r.t.} & \boldsymbol{w} = (w_1, \ldots, w_n),\ \sigma,\ \theta_+,\ \theta_-, \\ & \xi_1^+, \ldots, \xi_m^+,\ \text{and } \xi_1^-, \ldots, \xi_m^-, \\ \text{s.t.} & (\boldsymbol{w}, \sigma) \cdot (\boldsymbol{x}_i, y_i + \epsilon) \geq \theta_+ - \xi_i^+, \\ & (\boldsymbol{w}, \sigma) \cdot (\boldsymbol{x}_i, y_i - \epsilon) \leq \theta_- + \xi_i^-, \\ & \xi_i^+ \geq 0, \quad \text{and} \quad \xi_i^- \geq 0. \end{array}$$

Now it is easy to see that (P2*) can be solved similarly by using our OtpSoftMargin algorithm for (P2), within the following time bound.

Theorem 5.2 *Supposed that k $(= 1/D)$ is large enough so that two classes of shifted examples of* (P2*) *are linearly separable. Then we can solve* (P2*) *with the OptSoftMargin algorithm, and the average number of iterations executed is bounded by* $6\delta \ln m = O(n \ln m)$, *where* $|X| = m$ *and* $\delta \leq n + 2 + 2k$.

Let $(\boldsymbol{w}^*, \sigma^*, \theta_+^*, \theta_-^*)$ be a solution for (P2*). Again as in the HardTubeSVM, we can consider the ϵ^*-tube with $\epsilon^* = \epsilon - (\theta_+^* - \theta_-^*)/(2\sigma^*)$ (see Theorem 3.2 [9]). But this means here that all *normal* examples fit in this soft tube, while the outliers lie its outside.

6 Procedure for Determining the Parameter k

So far we have discussed by assuming that the parameter k is large enough so that the set of composed examples Z made up from X is linearly separable. But then, the choice of the parameter k becomes crucial, and we should provide some way to determine it. Here we give a way to decide whether k is sufficiently large, and propose some algorithms for choosing k.

Indeed, other formulations of SVMs always have at least one parameter that has to be tuned separately. In fact, changing the formulation into an equivalent one (like we have done for the Bennett and Bredensteiner case) does not always provide values for those parameters unless the correct solution is known. Thus, a procedure for determining a valid value for k remains necessary. In fact, as far as we know, very little is known about methods to choose the right values of these parameters in various formulations: the most standard way [16] is to try several choices and use the one with the best performance on the training set, using some sort of cross-validation mechanism.

Since we have assumed that k is chosen so that the set of composed examples is linearly separable, let us first consider the "linear separability" as a criteria for choosing k. We do not know, however, whether this is an appropriate criteria for k.

We hope that our discussion here would open up some algorithmic approach for determining the influence of erroneous examples.

In the context of linear programming type classification, Bennet and Mangasarian [7] proposed to use this "linear separability" for choosing weights that are similar to our influence parameter. There they provide a way to decide whether examples are linearly separable under a given choice of weights. Here we prove that a similar way works in our situation. Technically, we prove the following characterization.

Theorem 6.1 *Let X be any set of examples, and for any k, let Z be the set of composed examples made up from k examples from X. Let $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \xi^*)$ be a solution of* (P2) *on X. Then $\|\boldsymbol{w}^*\| > 0$ (or, equivalently $\boldsymbol{w}^* \neq \boldsymbol{0}$) if and only if k is large enough so that Z is linearly separable.*

Remark The problem (P5) does not make sense if $k > m_+$ or $k > m_-$, where m_+ and m_- are respectively the total number of positive and negative examples in X. Similarly, the problem (P2) is unbounded if $D < 1/m_+$ or $D < 1/m_-$. Thus, we assume below that $k \leq m_+$ and $k \leq m_-$.

First we note the following facts.

Fact 1 *Let* $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \xi^*)$ *be an optimal solution of* (P2). *Then the number of positive, respectively negative outliers w.r.t.* $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*)$ *is at most* k.

Proof The optimal solution $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \xi^*)$ satisfies the KKT-condition described in the proof of Lemma 3.2 (b). That is, the following holds for some $\boldsymbol{s}^*$ and $\boldsymbol{u}^*$. (Recall that P and N are used respectively to denote the set of indices of positive and negative examples. We use $\boldsymbol{e}$ to denote the vector with 1 at every entry.)

$$\boldsymbol{w}^* - \sum_{i \in P} s_i \boldsymbol{x}_i + \sum_{i \in N} s_i^* \boldsymbol{x}_i = 0, \qquad D\boldsymbol{e} - \boldsymbol{s}^* - \boldsymbol{u}^* = 0,$$

$$-1 + \sum_{i \in P} s_i^* = 0, \qquad -1 + \sum_{i \in N} s_i^* = 0,$$

$$\forall i \in P \ [s_i^*(\boldsymbol{w}^* \cdot \boldsymbol{x}_i - \theta_+^* + \xi_i^*) = 0], \qquad \forall i \in N \ [s_i^*(\boldsymbol{w}^* \cdot \boldsymbol{x}_i - \theta_-^* - \xi_i^*) = 0],$$

$$\boldsymbol{u}^* \cdot \xi^* = 0 \text{ (which means } (D\boldsymbol{e} - \boldsymbol{s}^*) \cdot \xi^* = 0), \text{ and } \xi^*, \boldsymbol{u}^*, \boldsymbol{s}^* \geq 0.$$

Now outliers are those $\boldsymbol{x}_i$ with $\xi_i^* > 0$. Then from the above, we have $D - s_i = 0$, i.e., $s_i = 1/k$, for every outlier $\boldsymbol{x}_i$. On the other hand, $\boldsymbol{s}$ must satisfy $\sum_{i \in P} s_i^* = \sum_{i \in N} s_i^* = 1$. Thus, there are at most k outliers among positive and negative examples respectively. □

Fact 2 *Let* $(\boldsymbol{w}, \theta_+, \theta_-, \boldsymbol{\xi})$ *be any feasible solution of* (P2) *such that* $\boldsymbol{w} = \boldsymbol{0}$. *Then the objective value corresponding to this solution is more than or equal to* 0.

Proof The objective value is clearly positive if $\theta_+ \leq \theta_-$. Thus, we consider here the case that $\theta_+ > \theta_-$.

Since $\boldsymbol{w} = \boldsymbol{0}$, it follows from the constraints of (P2) that $\theta_+ \leq \xi_i$ for every positive $\boldsymbol{x}_i$, and $-\theta_- \leq \xi_i$ for every negative $\boldsymbol{x}_i$. Then by using the assumption (see the remark of Theorem 6.1) that $Dm_+ \geq 1$ and $Dm_- \geq 1$, we have

$$\theta_+ \leq \min_{i:\, y_i = +1} \xi_i \leq D \cdot m_+ \cdot \left(\min_{i:\, y_i = +1} \xi_i \right) \leq D \cdot \sum_{i:\, y_i = +1} \xi_i, \quad \text{and} \tag{6}$$

$$-\theta_- \leq \min_{i:\, y_i = -1} \xi_i \leq D \cdot m_- \cdot \left(\min_{i:\, y_i = -1} \xi_i \right) \leq D \cdot \sum_{i:\, y_i = -1} \xi_i. \tag{7}$$

Hence, by adding both sides, we have $\theta_+ - \theta_- \leq D \cdot \sum_i \xi_i$; therefore, the objective value of (P2) for this solution is nonnegative. □

Proof of Theorem 6.1 Let Z_+ and Z_- respectively denote the set of all positive and negative composed examples of Z. By "Z is linearly separable", we formally mean that the following holds for *some* $\boldsymbol{w}$.

$$\min_{z_I \in Z_+} \boldsymbol{w} \cdot \boldsymbol{z}_I > \max_{z_J \in Z_-} \boldsymbol{w} \cdot \boldsymbol{z}_J. \tag{8}$$

(*The if direction*) Since Z is separable, there must be some $\boldsymbol{w}$ satisfying (8). Clearly $\boldsymbol{w} \neq \boldsymbol{0}$. We construct a feasible solution from $\boldsymbol{w}$ with the negative objective value, which, with Fact 2, proves that $\boldsymbol{w}^* \neq \boldsymbol{0}$.

First note that for any θ_+ and θ_-, there exists some $\boldsymbol{\xi}$ such that $(\boldsymbol{w}, \theta_+, \theta_-, \boldsymbol{\xi})$ satisfies the constraints of (P2). In particular, we can define θ_+ and θ_- so that

(a) $\ell_+ \leq k-1$, and there are $k - \ell_+$ positive examples $\boldsymbol{x}_i$ such that $\boldsymbol{w} \cdot \boldsymbol{x}_i = \theta_+$; and
(b) $\ell_- \leq k-1$, and there are $k - \ell_-$ negative examples $\boldsymbol{x}_i$ such that $\boldsymbol{w} \cdot \boldsymbol{x}_i = \theta_-$,

where ℓ_+ and ℓ_- are respectively the number of positive and negative examples $\boldsymbol{x}_i$ such that $\xi_i > 0$. (We also use $X_{\mathrm{err},+}$ and $X_{\mathrm{err},-}$ to denote respectively the set of positive and negative examples $\boldsymbol{x}_i$ such that $\xi_i > 0$.)

Then we have

$$\begin{aligned}\min_{z_I \in Z_+} \boldsymbol{w} \cdot \boldsymbol{z}_I &\leq \frac{(k-\ell_+)\theta_+ + \sum_{i \in X_{\mathrm{err},+}} (\theta_+ - \xi_i)}{k} \\ &= \theta_+ - \frac{1}{k} \cdot \sum_{i \in X_{\mathrm{err},+}} \xi_i = \theta_+ - D \cdot \sum_{i \in X_{\mathrm{err},+}} \xi_i.\end{aligned}$$

Similarly, we have

$$\max_{z_J \in Z_-} \boldsymbol{w} \cdot \boldsymbol{z}_J \geq \theta_- + D \cdot \sum_{i \in X_{\mathrm{err},-}} \xi_i.$$

Then it follows from (8) that

$$\theta_+ - D \cdot \sum_{i \in X_{\mathrm{err},+}} \xi_i > \theta_- + D \cdot \sum_{i \in X_{\mathrm{err},+}} \xi_i,$$

and hence,

$$(\theta_+ - \theta_-) - D \cdot \sum_i \xi_i \ > \ 0.$$

Now since the above value is not zero, there must be some $b > 0$ that satisfies

$$(\theta_+ - \theta_-) - D \cdot \sum_i \xi_i = b \|\boldsymbol{w}\|^2.$$

On the other hand, it is clear that $(a\boldsymbol{w}, a\theta_+, a\theta_-, a\boldsymbol{\xi})$ is also feasible for any $a > 0$, and the objective value of this solution is now computed as

$$\frac{1}{2}\|a\boldsymbol{w}\|^2 - (a\theta_+ - a\theta_-) + D \cdot \sum_i a\xi_i = \frac{a^2}{2}\|\boldsymbol{w}\|^2 - ab\|\boldsymbol{w}\|^2 = a^2\|\boldsymbol{w}\|^2\left(\frac{1}{2} - \frac{b}{a}\right).$$

Thus, by choosing a appropriately, we can make the objective value negative.

(*The only-if direction*) Suppose that an optimal solution $(\boldsymbol{w}^*, \theta_+^*, \theta_-^*, \xi^*)$ of (P2) satisfies $\boldsymbol{w}^* \neq 0$. We show that $\min_{z_I \in Z_+} \boldsymbol{w}^* \cdot \boldsymbol{z}_I > \max_{z_J \in Z_-} \boldsymbol{w}^* \cdot \boldsymbol{z}_J$.

Consider a composed example $\boldsymbol{z}_+$ with the smallest $\boldsymbol{w}^* \cdot \boldsymbol{z}_+$. Note that for any outlier $\boldsymbol{x}_i$ and any normal example $\boldsymbol{x}_j$, we have $\boldsymbol{w}^* \cdot \boldsymbol{x}_i < \boldsymbol{w}^* \cdot \boldsymbol{x}_j$. Furthermore, the number of positive outliers ℓ_+ is at most k (Fact 1). Hence, the value $\boldsymbol{w}^* \cdot \boldsymbol{z}_+$ is minimized if z_+ contains all positive outliers and the other (normal) elements $\boldsymbol{x}_j$ of z_+ satisfies $\boldsymbol{w}^* \cdot \boldsymbol{x}_j = \theta_+^*$. Thus, letting $X_{\mathrm{err},+}$ be the set of positive outliers, we have

$$\begin{aligned}\min_{z_+ \in Z_+} \boldsymbol{w}^* \cdot \boldsymbol{z}_+ &\geq \frac{(k - \ell_+)\theta_+^* + \sum_{\boldsymbol{x}_i \in X_{\mathrm{err},+}} (\theta_+^* - \xi_i^*)}{k} \\ &= \theta_+^* - \frac{1}{k} \cdot \sum_{\boldsymbol{x}_i \in X_{\mathrm{err},+}} \xi_i^* = \theta_+^* - D \cdot \sum_{\boldsymbol{x}_i \in X_{\mathrm{err},+}} \xi_i^*.\end{aligned}$$

Similarly, letting $X_{\mathrm{err},-}$ be the set of negative outliers, we have

$$\max_{z_- \in Z_-} \boldsymbol{w}^* \cdot \boldsymbol{z}_- \leq \theta_-^* + D \cdot \sum_{\boldsymbol{x}_i \in X_{\mathrm{err},-}} \xi_i^*.$$

Thus,

$$\min_{z_+ \in Z_+} \boldsymbol{w}^* \cdot \boldsymbol{z}_+ - \max_{z_- \in Z_-} \boldsymbol{w}^* \cdot \boldsymbol{z}_- \geq (\theta_+^* - \theta_-^*) - D \cdot \sum_i \xi_i^*.$$

On the other hand, since $\boldsymbol{w}^* \neq \boldsymbol{0}$ is an optimal solution and the objective value becomes 0 by the solution $(\boldsymbol{w}, \theta_+, \theta_-, \boldsymbol{\xi}) = (\boldsymbol{0}, 0, 0, \boldsymbol{0})$, the objective value corresponding to $\boldsymbol{w}^* \neq \boldsymbol{0}$ must be at most 0. Thus, we have

$$\frac{1}{2}\|\boldsymbol{w}^*\|^2 - (\theta_+^* - \theta_+^*) + D \cdot \sum_i \xi_i \leq 0,$$

which implies (noting $\|\boldsymbol{w}^*\|^2 > 0$)

$$(\theta_+^* - \theta_-^*) - D \cdot \sum_i \xi_i > 0.$$

Therefore, we have $\min_{z_I \in Z_+} \boldsymbol{w}^* \cdot \boldsymbol{z}_I > \max_{z_J \in Z_-} \boldsymbol{w}^* \cdot \boldsymbol{z}_J$. □

Assume, as our working hypothesis, that the separability of composed examples is an appropriate criterion for choosing the parameter k. Then this theorem provides us with an algorithmic way to determine k under this criterion. The simplest and straightforward way is to use the binary search method to find the smallest k $(= 1/D)$ such that (P2) has a solution with $\boldsymbol{w} \neq \boldsymbol{0}$. For this, we need to solve (P2) several times with different k, which may be computationally hard if there are huge number of examples. Here again our random sampling approach could be useful. Notice that k is not large enough even if we have $\boldsymbol{w} = \boldsymbol{0}$ for the local solution of (P2). Thus, one possible approach is to execute the algorithm of Fig. 3 with small k. If k is too small,

we may be able to find it in earlier steps, and then we can revise (e.g., double) k. Notice that we do not have to restart the algorithm; we can resume the computation with the revised k. By this way, we can combine the search for an appropriate choice of k with the random sampling iteration.

Notice that it is not necessary to use the same parameter k for all examples. In fact, it seems more reasonable [7] to use at least two independent parameters k_+ and k_- for positive and negative examples. Or we may be able to vary the influence parameter D $(= 1/k)$ on each example. For example, one can use small D_i for an example $\boldsymbol{x}_i$ that seems like an outlier. It would be very interesting if this adjustment of D_i's can be coordinated with the modification of the weight $u(\boldsymbol{x}_i)$ in our randomized sampling algorithm. This approach including theoretical and experimental investigations on the choice of D is left as our future work.

Acknowledgements We would like to thank Mr. Léonard Rodriguez for pointing out the problem in our earlier argument for estimating the combinatorial dimension of $(\mathcal{D}_2, \phi_2)$. Thanks are due to Emo Welzl for his help in understanding the applications of the Simple Sampling Lemma, to Jorge Castro for proofreading, and to Norbert Martinez for sharing some experimentation with us.

References

1. Adler, I., Shamir, R.: A randomized scheme for speeding up algorithms for linear and convex programming with high constraints-to-variable ratio. Math. Program. **61**, 39–52 (1993)
2. Balcázar, J.L., Dai, Y., Watanabe, O.: A Random sampling technique for training support vector machines: for a primal-form maximal-margin classifiers. In: Proceedings of 12th International Conference on Algorithmic Learning Theory (ALT'01). Lecture Notes in Computer Science, vol. 2225, pp. 119–134. Springer, London (2001)
3. Balcázar, J.L., Dai, Y., Watanabe, O.: Provably fast training algorithms for support vector machines. In: Proceedings of First IEEE International Conference on Data Mining (ICDM'01), pp. 43–50. IEEE, Los Alamitos (2001)
4. Balcázar, J.L., Dai, Y., Watanabe, O.: Provably fast support vector regression using random sampling. In: Proceedings of SIAM Workshop in Discrete Mathematics and Data Mining, pp. 19–29. SIAM, Philadelphia (2002)
5. Bennett, K.P., Bredensteiner, E.J.: Duality and geometry in SVM classifiers. In: Proceedings of 17th International Conference on Machine Learning (ICML'2000), pp. 57–64. Morgan Kaufmann, San Mateo (2000)
6. Bennett, K.P., Campbell, C.: Support Vector Machines: Hype or Hallelujah? SIGKDD Explorations **2**, 2 (2000)
7. Bennett, K.P., Mangasarian, O.L.: Robust linear programming discrimination of two linearly inseparable sets. Optim. Methods Softw. **1**, 23–34 (1992)
8. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (1995)
9. Bi, J., Bennett, K.P.: Duality, geometry, and support vector regression. In: Proceedings of Advances in Neural Information Processing Systems (NIPS'02), pp. 539–600. MIT Press, Cambridge (2002)
10. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of 5th Annual Conference on Computational Learning Theory (COLT'92), pp. 144–152. ACM, New York (1992)
11. Bradley, P.S., Mangasarian, O.L., Musicant, D.R.: Optimization methods in massive datasets. In: Abello, J., Pardalos, P.M., Resende, M.G.C. (eds.) Handbook of Massive Datasets, pp. 439–471. Kluwer Academic, Dordrecht (2002)
12. Cauwenberghs, G., Poggio, T.: Incremental and decremental support vector machine learning. In: Proceedings of Advances in Neural Information Processing Systems (NIPS'00), pp. 409–415. MIT Press, Cambridge (2000)
13. Clarkson, K.L.: A Las Vegas algorithm for linear programming when the dimension is small. In: Proceedings of 29th IEEE Symposium on Foundations of Computer Science (FOCS'88), pp. 452–456. IEEE, Los Alamitos (1988). (See [15] for a better version.)

14. Clarkson, K.L.: Las Vegas algorithms for linear and integer programming. J. ACM **42**, 488–499 (1995)
15. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**, 273–297 (1995)
16. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
17. Cucker, F., Smale, S.: On the mathematical foundations of learning. Bull. Am. Math. Soc. **39**(1), 1–49 (2001)
18. Evgeniou, T., Pontil, M., Poggio, T.: A unified framework for regularization networks and support vector machines. AI Memo No. 1654, AI Lab MIT (1999)
19. Fine, S., Scheinberg, K.: Efficient SVM training using low-rank kernel representations. J. Mach. Learn. Res. **2**, 243–264 (2001)
20. Gärtner, B., Welzl, E.: A simple sampling lemma: Analysis and applications in geometric optimization. Discrete Comput. Geom. **25**(4), 569–590 (2001)
21. Hush, D., Scovel, C.: Polynomial-time decomposition algorithms for support vector machines. Mach. Learn. **51**, 51–71 (2003)
22. Keerthi, S.S., Gilbert, E.G.: Convergence of a generalized SMO algorithm for SVM classifier design. Mach. Learn. **46**(1–3), 351–360 (2002)
23. Lee, Y., Mangasarian, O.L.: RSVM: Reduced Support Vector Machines. In: CD Proceedings of the SIAM International Conference on Data Mining, Chicago, 5–7 April 2001. SIAM, Philadelphia (2001). (Available from http://www.cs.wisc.edu/ olvi/olvi.html.)
24. Lin, C.J.: Linear convergence of a decomposition method for support vector machines. Technical Report. http://www.csie.ntu.edu.tw/~cjlin/papers/linearconv.pdf (2001)
25. Lin, C.J.: On the convergence of the decomposition method for support vector machines. IEEE Trans. Neural Networks **14**, 1267–1281 (2002)
26. Martín, M.: On-line support vector machine regression. In: Proceedings of 13th European Conference on Machine Learning (ECML'02), pp. 282–294 (2002).
27. Osuna, E., Freund, R., Girosi, F.: An improved training algorithm for support vector machines. In: Proceedings of IEEE Workshop on Neural Networks for Signal Processing, pp. 276–285 (1997)
28. Pavlov, D., Mao, J., Dom, B.: Scaling up support vector machines using boosting algorithm. In: Proceedings of International Conference Pattern Rec., pp. 2219–2222 (2000)
29. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods—Support Vector Learning, pp. 185–208. MIT Press, Cambridge (1999)
30. Schölkopf, B., Burges, C., Vapnik, V.: Extracting support data for a given task. In: Proceedings of First International Conference on Knowledge Discovery and Data Mining (KDD'95), pp. 252–257 (1995)
31. Schölkopf, B., Smola, A., Williamson, R.C.: A new support vector regression algorithm. In: Proceedings of Advances in Neural Information Processing Systems (NIPS'99), pp. 330–336. MIT Press, Cambridge (1999)
32. Schölkopf, B., Smola, A., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Comput. **12**, 1207–1245 (2000)
33. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London (1998)