Alcom-FT Technical Report Series
ALCOMFT-TR-02-84

Provably Fast Support Vector Regression Using Random Sampling*

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

Yang Dai
Dept. of Bioengineering (MC063)
University of Illinois at Chicago
851 S. Morgan Street, Chicago, IL 60607–7052, USA

Osamu Watanabe
Dept. of Mathematical and Computing Sciences,
Tokyo Institute of Technology
Meguro-ku Ookayama, Tokyo 152-8552, Japan

Abstract

Support Vector Machines are a family of data analysis algorithms, based on convex Quadratic Programming. Their use has been demonstrated in classification, regression, and clustering problems. In previous work we have proved that a random sampling technique based on an evolving discrete probability distribution provides a training algorithm for Support Vector Classification with provably low expected running time as a function of the number of points. This paper proves that the same technique can be applied to Support Vector Regression, by applying and reinterpreting an intuitive geometric breakthrough by Bi and Bennett, and obtaining, through the use of such discrete techniques, the first (to our knowledge) support vector regression algorithm with a provably low expected running time as a function of the number of examples. This is expected to be useful for the processing of extremely large datasets.

*The first author is supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT), EU EP27150 (Neurocolt II), Spanish Government PB98-0937-C04 (FRESCO), and CIRIT 1997SGR-00366. The second author is supported by a start-up fund from Dept. of Bioengineering, University of Illinois at Chicago. 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.

1. Introduction

The *Support Vector Machine* (SVM in short) is a modern mechanism for two-class classification, regression, and clustering problems. Since the present form of SVM was proposed [CV95], 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 [CST00]. 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 three main characteristics of SV algorithms are: first, that they implement the approach called Structural Risk Minimization to reduce the generalization error; second, that they work on high-dimensional feature spaces by means of a dual formulation in terms of kernels; and, third, that the prediction is based on hyperplanes in these feature spaces, which may correspond to quite involved surfaces on the input space, and which can handle outliers in the training set by means of soft margins or soft tubes.

Algorithmically, the problem amounts to solving a convex quadratic programming (QP in short) problem. Convex QP problems are polynomial-time solvable, but still not so easy. To scale up to really large data sets, the standard QP algorithms alone are inappropriate since their running times grow too fast (according to [BC00], even the very best methods are typically quadratic). Thus, many algorithms and implementation techniques have been developed for using SVMs efficiently; see, e.g., [SS98, BMM00].

Among the proposed speed-up techniques, those called "subset selection" [SS98] have been used as effective heuristics already from the earliest papers eventually leading to SVM, including [BGV92]. Roughly speaking, a *subset selection* is a technique to speed-up the algorithm by dividing the original QP problem into small pieces, thereby reducing the size of each QP problem. Well-known subset selection techniques are chunking, decomposition, and sequential minimal optimization (SMO in short), which may be combined with a reduction on the data on the basis of so-called guard vectors. (See [CV95, OFG97, Pla99, CST00, YA00] for the details.)

In previous papers [BDW01a], [BDW01b], we focused on algorithms for training two-class classification SVMs. We have proposed to use discrete random sampling techniques that have been developed and used for combinatorial optimization problems; see, e.g., [Cla88, AS93, GW00]. In particular, we have developed the analysis, based on the beautiful Simple Sampling Lemma of [GW00], of certain SVM training algorithms for classification. Specifically, the running time of the algorithm obtained is polynomial (with low degree for not too many outliers) in the dimension n of the input space and quasilinear, $O(m \log m)$, on the number m of data points, for m larger enough than n. This complexity-theoretic analysis suggests that its scalability properties to truly large data sets will be acceptable.

The contribution of this paper is a randomized subset selection scheme for a different problem, namely, Support Vector Regression (SVR), with respect to the so-called ϵ-insensitive loss usually employed for SVR. It is heavily based on the recent work [BB01], which describes and formally proves how the regression problem can be reduced to a classification problem. We briefly review this important contribution in section 2, together with some more preliminaries. It turns out that our previously developed techniques apply to the classification problem obtained; thus, we are left with the tasks of working out the details, which we explain intuitively in sections 3 and 4, and of reinterpreting in terms of regression the concepts employed to develop our approach in section 5. Two appendices provide the formalization of the algorithms and their analysis supporting the claims in the paper.

2. Regression, Classification, and Support Vectors Machines

The regression problem under ϵ-insensitive loss by means of SVR can be phrased as follows. Given is a set of points in a product space $\mathcal{X} \times \mathcal{Y}$ where the coordinates in $\mathcal{X}$ act as predictor (or given) information and the coordinate in $\mathcal{Y}$ act as response (or desired) variable. Given is also the value of ϵ for the loss function. We have to come up with 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 $\mathcal{Y}$ coordinate (thus all are within a so-called ϵ-tube around the hyperplane). Of course, if ϵ is too small, 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: soft margins become here slack quantities that allow, but penalize, points that are too far from the hyperplane.

As with SVM classification, some applications of the method to large datasets raise difficulties in handling huge quadratic programs. Again subset selection techniques have been applied to regression problems [SS98]. Our contribution is now yet another subset selection technique for which formal theorems can be proved bounding the expected time necessary for convergence.

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 responses), and $\boldsymbol{x}_i$ and y_i to denote the ith example and its response respectively. The range of i is always $\{1, ..., m\}$.

We briefly explain now the reduction of regression to classification suggested in [BB01]. We only describe now the hard tube case, and return to the general form in a later section.

There are several formalizations of the SVM regression problem; see the references in [BB01]. The one proposed in that paper, which we are following, 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, one constructs 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 a possible formalization for the problem is as follows. Suppose that we are given a set of m examples $\boldsymbol{x}_i$, $1 \le i \le m$, in some n dimensional space, say $\mathbb{R}^n$. 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. The *SV regression problem* we will discuss in this paper essentially consists in solving the optimization problem (P1) in Figure 1. We are assuming here that ϵ is large enough for the problem to have a solution.

The two bias parameters are not independent; this formulation conveniently follows [BB00] where a connection

SVR (P1)

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

Soft Tube SVR (P2)

$$\begin{aligned}
&\text{min. } \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2}\delta^2 - (\theta_+ - \theta_-)\\
&\quad + D \cdot (\textstyle\sum_i \xi_i + \sum_i \eta_i),\\
&\text{w.r.t. } \boldsymbol{w} = (w_1, ..., w_n),\ \delta,\ \theta_+,\ \theta_-,\\
&\quad \xi_1, ..., \xi_m, \text{ and } \eta_1, ..., \eta_m\\
&\text{s.t. } \boldsymbol{w} \cdot \boldsymbol{x}_i + \delta(y_i + \epsilon)\ \geq\ \theta_+ - \xi_i\\
&\text{and } \boldsymbol{w} \cdot \boldsymbol{x}_i + \delta(y_i - \epsilon)\ \leq\ \theta_- + \eta_i\\
&\text{and } \xi_i\ \geq\ 0\\
&\text{and } \eta_i\ \geq\ 0
\end{aligned}$$

Figure 1. Two Optimization Problems: Regression

Max Margin (P3)

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

Max Soft Margin (P4)

$$\begin{aligned}
&\text{min. } \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2}\delta^2 - (\theta_+ - \theta_-)\\
&\quad + D \cdot (\textstyle\sum_i \xi_i + \sum_i \eta_i),\\
&\text{w.r.t. } \boldsymbol{w} = (w_1, ..., w_n),\ \delta,\ \theta_+,\ \theta_-,\\
&\quad \xi_1, ..., \xi_m \text{ and } \eta_1, ..., \eta_m\\
&\text{s.t. } (\boldsymbol{w}, \delta) \cdot (\boldsymbol{x}_i, y_i + \epsilon)\ \geq\ \theta_+ - \xi_i\\
&\text{and } (\boldsymbol{w}, \delta) \cdot (\boldsymbol{x}_i, y_i - \epsilon)\ \leq\ \theta_- + \eta_i\\
&\text{and } \xi_i\ \geq\ 0\\
&\text{and } \eta_i\ \geq\ 0.
\end{aligned}$$

Figure 2. Two More Optimization Problems: Classification

to reduced convex hulls was developed that is crucial for our work.

In the same way that SVM classifiers are also applicable to the nonseparable case, an important feature of SVR is that it can be applied also for too small a value of ϵ. This is done in classification by relaxing constraints by introducing slack variables or "soft margin error", and the reduction of regression to classification provides a similar solution now. Thus, we also consider the generalization of the problem (P1), corresponding to the soft tube regression problem: it is (P2) in Figure 1. Problems (P1) and (P2) in Figure 1 are respectively problems (5) and (7) in [BB01].

By the *solution* of (P1), we mean the hyperplane that achieves the minimum cost. Given a solution, its support vectors are the data points $\boldsymbol{x}_i$ for which, at the solution, at least one of the corresponding inequalities is tight: $\boldsymbol{w} \cdot \boldsymbol{x}_i + \delta(y_i + \epsilon) = \theta_+$ or $\boldsymbol{w} \cdot \boldsymbol{x}_i + \delta(y_i - \epsilon) = \theta_-$. We also consider partial problems of (P1) that minimize 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.

For a given set X of examples, suppose we solve the problem (P2) and obtain the optimal hyperplane. Then an example $\boldsymbol{x}_i \in X$ is called an *outlier* if its response variable is off the value predicted by the hyperplane by more than ϵ. Throughout this paper, we use ℓ to denote the number of outliers. The softness parameter D 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.

Now, the lowest points in the original data become the support vectors above the hyperplane, whereas the highest points become support vectors below it (all the relative comparisons are in terms of the response variable coordinate). Using this hyperplane as regressor presents a number of advantages argued in [BB01] with respect to other existing formulations. A number of facts are derived there from the Wolfe dual formulation of the primal problem (P1). The main result is that, in dual space, one can see the dual variables as pointing out to two points, one within each of the two convex hulls determined respectively by the points shifted up and the points shifted down, and the optimization as finding the closest points in these convex hulls. The hyperplane is obtained by bisecting the segment joining those two points.

A slight adjustment of (P1) gives an equivalent specific classification problem as in (P3) in Figure 2, where we can see that our examples have both their predictor and response coordinates, and are considered as positive examples in the $+\epsilon$ case and as negative examples in the $-\epsilon$ case. We will also use later on the problem (P4) in Figure 2, which will correspond to the soft tubes case (P2).

3. Hard Tubes

According to the reduction explained in the previous section, this boils down to the separable case (P3) of classification. We can solve this optimization problem by using the algorithm in [BDW01a] for training support vector classifiers.

The idea is simple. Pick a certain number of examples from X (polynomial in n and independent of m) 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 some examples again randomly according to their weights. If we iterate this process several rounds, the weight of "important examples", which are the support vectors of the hard margin classification problem (P3), 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 global one, which is detected since all points are correctly classified and none forces smaller margin; the algorithm terminates at this point. By using Welzl's and Gaertner's Simple Sampling Lemma, given in Appendix A, we proved in [BDW01a] that the algorithm terminates in $O(n \log m)$ rounds on average. For completeness, we include too in appendix A the precise algorithm and the proof of its running time.

By applying this algorithm to the two sets of points obtained through the $+\epsilon$ and $-\epsilon$ y-translations of our input, we are actually solving the classification problem corresponding to the transformation given by Bi and Bennett, and therefore we obtain a solution to the regression problem.

More precisely, we obtain a solution in a more general sense since we not only have now the hyperplane, but also the two biases that indicate both borders (up and down) of the solution hard tube; thus we are fitting the data up to a possibly smaller value of ϵ, namely $\hat{\epsilon} = \epsilon - (\hat{\theta}_+ - \hat{\theta}_-)/\hat{\delta}$, for the optimal values $\hat{\theta}_+$, $\hat{\theta}_-$, and $\hat{\delta}$ of (P3). See Theorem 3.1 of [BB01].

4. Soft Tubes

Now we move to the generalized form of reduction of regression to classification from [BB01], which in turn is based on the geometric interpretation (by reduced convex hulls) of soft margin classification given in [BB00] after the Wolfe dual of (P2). Essentially, the way of stating the Wolfe dual can be interpreted as follows: find as before the closest pair of points, one on each of the two areas obtained by shifting up and down the data, but disallowing the use of the most risky parts of the convex hulls. Indeed, for too small ϵ there is no solution to the regression problem, and this manifests itself as intersecting convex hulls in the dual formulation. Then, reduced convex hulls are used instead, where each point of the reduced convex hull has to be obtained through the contribution of at least k points. This allows one to discard individual outliers. The parameter k has to be large enough so that the reduced convex hulls do not intersect, and is related to the balance parameter C in the standard soft margin SVM classification approach. The formal statement that reduced convex hulls indeed lead to the Wolfe dual problem of the soft tube case can be found in [BB01].

We proposed and analyzed in [BDW01a] and in [BDW01b] algorithms for the soft margin classification case. In particular, we use the second one here, since it is the only one for which the complete analysis is available. There, we recasted the reduced convex hulls approach in the form of composed examples, and used them to develop the analysis of the algorithm.

The algorithm itself is pretty much like the hard margin one, with the same resampling and doubling the probability of those points outside the local solution. The only essential difficulty is in constructing a stopping criterion, since the one used for the hard tube, that of having all points correctly classified with large enough margin, will never occur.

The analysis of the algorithm in [BDW01b] proves first that, as one is looking into separable reduced convex hulls, the problem is indeed the same as the previous one on the so-called composed examples; unfortunately they are too many to test and would slow down too much the algorithm. Then the main result of [BDW01b] proves that it is enough to test for closeness to the local solution exactly those original points that were left unsampled at the current round.

Thus the algorithm consists again in sampling, constructing a local classifier separating the $+\epsilon$ from the $-\epsilon$ points, computing the margin by appeal to a technical lemma, and doubling the weight of those points that were not sampled and remain either to the wrong side of the classifying hyperplane, or closer than the margin computed. For completeness we present the algorithm and a hint of its analysis in Appendix B, where also we prove a new lemma regarding the support vectors among the composed examples.

5. Reinterpreting the Algorithm as Regression

A drawback of the previous algorithm is that, being based on classification, it obscures what is really going on in terms of regression. We recast the algorithm now fully in terms of regression, and indicate how a technical lemma given in Appendix B allows us also to thus obtain additional explanations regarding the value $\hat{\epsilon}$ obtained.

Speaking only in terms of the regression problems in Figure 1, we can explain our algorithm as follows: first, initialize all weights to 1 and choose the appropriate sample size r; then repeat: sample r points according to the current weights and compute the soft tube regressor for those points, by solving problem (P2); from the optimal solution, compute the tube radius $\hat{\epsilon} = \epsilon - (\hat{\theta}_+ - \hat{\theta}_-)/\hat{\delta}$; if all unsampled points are within the tube, which means at distance less than $\hat{\epsilon}$ of the solution hyperplane, then stop; otherwise, if there are not too many unsampled points outside the tube, double their weights.

The lemmas in [BDW01b], together with the new lemma in Appendix B, provide the right value of the sample size r, which is independent on the number of input data points, and guarantee that the algorithm stops precisely when the global solution has been found, and that it is expected to stop within a number of trials that is logarithmic in m.

6. Conclusions

We have extended algorithms for training support vector machines in classification problems, previously developed, into algorithms for Support Vector Regression problems under ϵ-insensitive loss. The connection was possible using the reduction of hard tubes into separable classification, and soft tubes into soft margin classification, given in [BB01]. In fact, the soft tube case was handled by reducing it into the hard tube case through reduced convex hulls or, in our case, through composed examples.

The main properties of our algorithm, inherited from the SVM classifier training algorithms, are that we can formally prove rigorous fast convergence results, in the common case of having larger data sets than the dimension. Our main algorithm has been formally proved to have a very mild dependence of the number of data points, although its dependence on the number of outliers for the soft tubes case may become a cause of slow computation. It needs the prior knowledge of a parameter k corresponding to the influence of outliers: see [BB01] for a discussion of the advantages of this parameterization compared to other free parameters in alternative formulations.

Additionally, by analyzing the KKT conditions, we have proved a lemma that explains exactly how the support vectors among the composed examples, which we had to introduce for technical reasons, can be interpreted in terms of the original regression problem with no direct reference to the reduction to classification or to the reduced convex hulls dual problems.

As for experimentation, we do not have relevant results yet. As happened with classification, we do not expect a naive implementation of our algorithm to be competitive immediately. A number of implementation problems to be solved for the classification case were enumerated in [BDW01b]; they remain to be solved as well for the regression case.

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.

[BDW01a] J. L. Balcázar, Y. Dai, and O. Watanabe, A random sampling technique for training support vector machines, in *Proc. Algorithmic Learning Theory* (ALT'01), 2001.

[BDW01b] J. L. Balcázar, Y. Dai, and O. Watanabe, Provably fast training algorithms for support vector machines, in *Proc. First IEEE International Conference on Data Mining* (ICDM'01), 2001.

[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.

[BC00] K. P. Bennett and C. Campbell, Support Vector Machines: Hype or Hallelujah?, SIGKDD *Explorations* Newsletter 2, 2 (2000).

[BB01] J. Bi and K. P. Bennett, Duality, geometry, and support vector regression, in Advances in Neural Information Processing Systems, NIPS*2001 (to appear in MIT Press, 2002; available from http://www.rpi.edu/ bennek/).

[Ber95] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik, A training algorithm for optimal margin classifiers, in *Proc. Int. Conf. on Computational Learning Theory* (COLT'92), 144–152, 1992.

[BMM00] P. S. Bradley, O. L. Mangasarian, and D. R. Musicant, Optimization methods in massive datasets, in *Handbook of Massive Datasets* (J. Abello , P. M. Pardalos, and M. G. C. Resende, eds.) Kluwer Academic Pub., 2001, to appear.

[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.

[CST00] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press 2000.

[GW00] B. Gärtner and E. Welzl, A simple sampling lemma: Analysis and applications in geometric optimization, *Discr. Comput. Geometry*, 2000, to appear. (Also available from http://www.inf.ethz.ch/personal /gaertner/publications.html.)

[OFG97] E. Osuna, R. Freund, and F. Girosi, An improved training algorithm for support vector machines, in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, 276−285, 1997.

[Pla99] J. Platt, Fast training of support vector machines using sequential minimal optimization, in *Advances in Kernel Methods – Support Vector Learning* (B. Scholkopf, C. Burges, and A. J. Smola, eds.), MIT Press, 185−208, 1999.

[SS98] A. J. Smola and B. Schölkopf, A tutorial on support vector regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, 1998.

[YA00] M.-H. Yang and N. Ahuja, A geometric approach to train support vector machines, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, 430–437.

7. Appendix A

We explain now, briefly, the essentials of the abstract framework for discussing randomized sampling techniques that was given by Gärtner and Welzl [GW00]. Then we apply it for the analysis of the hard tube algorithm.

Randomized sampling techniques, particularly, the Sampling Lemma below, are applicable for many "LP-type" problems. LP stands for Linear Programming. 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 regard $\mathcal{D}$ as X and define ϕ as a mapping from a given subset R of X to the local solution of (P1) for the subset of constraints corresponding to R. As an LP-type problem, we require $(\mathcal{D}, \phi)$ to satisfy certain conditions. Here we omit the explanation and simply mention that our example case 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 $\mathcal{D}$. For the problem (P1), each basis is a minimal set of support vectors. The combinatorial dimension of (P1) is $n+2$ since the two bias parameters θ_+ and θ_- are not independent.

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 R of X, let $(\boldsymbol{w}, \delta, \theta_+, \theta_-)$ be a local solution of (P1) obtained for R. Then $\boldsymbol{x}_i \in X$ is a *violator* of R, or (more directly) a *violator* of $(\boldsymbol{w}, \delta, \theta_+, \theta_-)$, if the constraint corresponding to $\boldsymbol{x}_i$ is not satisfied by $(\boldsymbol{w}, \delta, \theta_+, \theta_-)$.

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*, a set containing some element more than once. In order to discuss the case where 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.

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

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.

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

We are ready to state the Sampling Lemma. The idea, and its algorithmic application through this appendix, can be found already in the paper by Clarkson [Cla88], where a randomized algorithm for linear programming has been proposed. Indeed a similar idea has been used [AS93] to design an efficient randomized algorithm for quadratic programming. Here we explain the framework and the algorithm following [GW00].

Lemma 1 *Let $(\mathcal{D}, \phi)$ be any LP-type problem. Let d be its combinatorial dimension. 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 $v_{\mathcal{R}}$:*

$$\mathrm{Exp}(v_{\mathcal{R}}) \;\leq\; \frac{u(\mathcal{D}) - r}{r+1} \cdot d \qquad (3)$$

See [GW00] for the proof, additional explanations, variations for other sampling schemas, important related results such as tail bounds, and a large number of incarnations of this Sampling Lemma.

Now we are in a position to describe more precisely and analyze the algorithm in section 3, taken from [BDW01a]. We use u there 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

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

Figure 3. Algorithm for the Hard Case

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 1. From it, we can prove the following bound.

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

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 $u(V) \leq u(X)/3(n+2)$, by doubling the weight of all examples in V, i.e., all violators, $u(X)$ increases by at most $u(X)/3(n+2)$. Since $u(X)$ is initially m, after t successful iterations, we have $u(X) \leq m(1 + 1/(3(n+2)))^t$.

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 R, i.e., $X_0 \subseteq R$, then there can be no violator for R. At each successful iteration (if it is not the end) some $\boldsymbol{x}_i$ of X_0 must not be in R, which implies that it is a violator of R by the fact that X_0 is a basis. Hence, $u(\boldsymbol{x}_i)$ gets doubled. Since $|X_0| \leq (n+2)$, there is some $\boldsymbol{x}_i$ in X_0 that gets doubled at least once every $n+1$ successful iterations. Therefore, after t successful iterations, $u(\boldsymbol{x}_i) \geq 2^{t/n+2}$.

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

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

This implies that $t < 3(n+2)\ln m$ as long as we have not reached the last iteration. That is, the algorithm terminates within less than $3(n+2)\ln m$ 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 R. Then this R corresponds to $\mathcal{R}$ in the Sampling Lemma, and we have $u(V) = v_{\mathcal{R}}$. Hence, from the bound in Lemma 1, we can bound the expectation of $u(V)$ by $(u(X) - r)(n+2)/(r+1)$, which is smaller than $u(X)/(6(n+2))$ 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 $2 \cdot 3(n+2)\ln m$ steps. □

Thus, while this 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, then this algorithm is faster than solving (P1) directly.

8. Appendix B

Let Z be the set of *composed examples* $\boldsymbol{z}_I$ that are defined as $\boldsymbol{z}_I = (\boldsymbol{x}_{i_1} + \boldsymbol{x}_{i_2} + \cdots + \boldsymbol{x}_{i_k})/k$, with some k distinct elements $\boldsymbol{x}_{i_1}$, $\boldsymbol{x}_{i_2}$, ..., $\boldsymbol{x}_{i_k}$ of $\mathcal{X} \times \mathcal{Y}$. We use I for indexing elements of Z. Note that each composed example is an average of data points, including the response variable. The range of I is $\{1, ..., M\}$, where $M \stackrel{\text{def}}{=} |Z|$; each such I can be identified with a set $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$. 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. Note also that these composed examples are sort of mass centers of all groups of k data points.

Next, we shift up and down Z by adding and subtracting, respectively, ϵ to the $\mathcal{Y}$ coordinate. Then the resulting sets of composed examples, $\boldsymbol{z}_I^+$ and $\boldsymbol{z}_I^-$, for large enough k, become linearly separable. We can formulate now:

Max Margin for Composed Examples (P5)

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

Problem (P5) which is exactly like problem (P1) on the composed examples from Z. Let $(\boldsymbol{w}^*, \delta^*, \theta_+^*, \theta_-^*)$ and $(\boldsymbol{w}^*, \delta^*, \eta_+^*, \eta_-^*)$ be the solutions of (P2) for X and (P5) for Z respectively: they share $(\boldsymbol{w}^*, \delta^*)$ as proved in [BB00].

This situation is suitable for the sampling technique, and the same algorithm as in Appendix A could be applied. However, it would need to keep weights of all M composed examples in Z and, for finding violators and for modifying weights, we have to go through Z, which takes at least $O(M)$ steps. Partial solutions to these problems were given

procedure OptSoftMargin
set weight $u(\boldsymbol{x}_i)$ to be 1 for all examples in X;
$r \leftarrow 6k(n+2)^2$;
repeat
$R \leftarrow$ choose r examples from X
randomly according to u;
let $(\boldsymbol{w}, \delta, \theta_+, \theta_-)$ be a solution of (P2) for R;
$V \leftarrow$ the set of violators in $X - R$ of the solution;
if $u(V) \leq u(X)/(3k(n+2))$ **then**
double the weight $u(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i \in V$;
until $V = \emptyset$;
return the last solution;
end-procedure.

Figure 4. Algorithm for the Soft Case

in [BDW01a] and an alternative complete solution was provided in [BDW01b]. There we proved that it was equivalent to test composed examples or to test original examples if done in the appropriate way. The resulting algorithm appears in Figure 4, and we know from [BDW01b] that a theorem analogous to Theorem 2 holds for it.

The main observation now is that the composed examples that would be support vectors of the hard tube fulfill the following properties:

Lemma 3 *Consider the composed support vectors of problem (P1) for the composed examples Z. Then, among the initial data points,*

1. *all outliers belong to all the composed support vectors, and*
2. *all not-outlier points in the composed support vectors lie on the border of the soft tube, at distance $\hat{\epsilon}$ of the regression hyperplane;*
3. *all other points are at distance at most $\hat{\epsilon}$ of the regression hyperplane, thus also within the soft tube.*

Indeed, for a given example set X consisting of $(\boldsymbol{x}_i, y_i + \epsilon)$ and $(\boldsymbol{x}_i, y_i - \epsilon)$, let Z be the set of composed examples. Consider problem (P5), and solutions $(\boldsymbol{w}^*, \delta^*, \theta_+^*, \theta_-^*)$ and $(\boldsymbol{w}^*, \delta^*, \eta_+^*, \eta_-^*)$ of (P2) for X and (P5) for Z respectively.

Let $X_{\mathrm{err},+}$ and $X_{\mathrm{err},-}$ denote respectively the sets of positive and negative outliers. That is, $(\boldsymbol{x}_i, y_i + \epsilon)$ (resp. $(\boldsymbol{x}_i, y_i - \epsilon)$) belongs to $X_{\mathrm{err},+}$ (resp., $X_{\mathrm{err},-}$) if and only if $(\boldsymbol{w}^*, \delta^*) \cdot (\boldsymbol{x}_i, y_i + \epsilon) < \theta_+^*$ (resp., $(\boldsymbol{w}^*, \delta^*) \cdot (\boldsymbol{x}_i, y_i - \epsilon) > \theta_-^*$). We use ℓ_+ and ℓ_- to denote respectively the number of positive and negative outliers. Recall that we assume that our constant k is larger than both ℓ_+ and ℓ_-, and that in fact problem (P5) is linearly separable. Let $X_{\mathrm{err}} = X_{\mathrm{err},+} \cup X_{\mathrm{err},-}$.

The problem (P5) is regarded as the LP-type problem $(\mathcal{D}, \phi)$, where the correspondence is the same as (P1) except that Z is used as $\mathcal{D}$ here. Let Z_0 be a basis 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), in [BDW01a] we showed a lemma for such classification cases that, translated into our regression setting, reads:

Lemma 4 *Let $\boldsymbol{z}_I$ be any positive final extremer. Then the following properties hold: (a) $(\boldsymbol{w}^*, \delta^*) \cdot \boldsymbol{z}_I = \eta_+^*$. (b) $X_{\mathrm{err},+} \subseteq z_I$. (c) For every $(\boldsymbol{x}_i, y_i + \epsilon) \in z_I$, if $(\boldsymbol{x}_i, y_i + \epsilon) \notin X_{\mathrm{err},+}$, then we have $(\boldsymbol{w}^*, \delta^*) \cdot (\boldsymbol{x}_i, y_i + \epsilon) = \theta_+^*$.*

The corresponding facts hold, *mutatis mutandis*, for negative final extremers. ¿From this lemma, we complete the proof of our new lemma as follows. The fact (a) of the lemma shows that all final extremers are located on some hyperplane whose distance from the base hyperplane $(\boldsymbol{w}^*, \delta^*) \cdot \boldsymbol{z} = 0$ is η_+^*. Essentially, (b) means that, when the final extremers are obtained, the set of outliers can be obtained as $\cap_{\boldsymbol{z}_I \in Z_0} z_I$, the set of examples appearing in all final extremers. On the other hand, the fact (c) states that all *normal* examples in X (i.e., examples not in $X_{\mathrm{err},+}$, each final extremer has $k - \ell_+$ of them) appearing in some final extremer are all located again on some hyperplane whose distance from the base hyperplane is $\theta_+^* > \eta_+^*$. From $(\boldsymbol{w}^*, \delta^*) \cdot (\boldsymbol{x}_i, y_i + \epsilon) = \theta_+^*$, (resp.$(\boldsymbol{w}^*, \delta^*) \cdot (\boldsymbol{x}_i, y_i - \epsilon) = \theta_-^*$,) we have

$$y_i - \left(\frac{-\boldsymbol{w}^*}{\delta^*}\right)\boldsymbol{x}_i - \frac{\theta_+^* + \theta_-^*}{2\delta^*} = -\epsilon + \frac{\theta_+^* - \theta_-^*}{2\delta^*}$$

$$(\text{resp.} y_i - \left(\frac{-\boldsymbol{w}^*}{\delta^*}\right)\boldsymbol{x}_i - \frac{\theta_+^* + \theta_-^*}{2\delta^*} = \epsilon - \frac{\theta_+^* - \theta_-^*}{2\delta^*})$$

This means that they lie on the boundary of the $(\epsilon - \frac{\theta_+^* - \theta_-^*}{2\delta^*})$-tube. On the other hand, for any point in $X_{\mathrm{err},+}$ (resp. $X_{\mathrm{err},-}$) we have $(\boldsymbol{w}^*, \delta^*) \cdot (\boldsymbol{x}_i, y_i + \epsilon) < \theta_+^*$, (resp.$(\boldsymbol{w}^*, \delta^*) \cdot (\boldsymbol{x}_i, y_i - \epsilon) > \theta_-^*$). This means that the outliers lie outside the $(\epsilon - \frac{\theta_+^* - \theta_-^*}{2\delta^*})$-tube.