Tight bounds for SVM classification error

B. Apolloni, S. Bassis, S. Gaito, D. Malchiodi
Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
Via Comelico 39/41, 20135 Milano, Italy
E-mail: {apolloni, bassis, gaito, malchiodi}@dsi.unimi.it

Abstract— **We find very tight bounds on the accuracy of a Support Vector Machine classification error within the Algorithmic Inference framework. The framework is specially suitable for this kind of classifier since (i) we know the number of support vectors really employed, as an ancillary output of the learning procedure, and (ii) we can appreciate confidence intervals of misclassifying probability exactly in function of the cardinality of these vectors. As a result we obtain confidence intervals that are up to an order narrower than those supplied in the literature, having a slight different meaning due to the different approach they come from, but the same operational function. We numerically check the covering of these intervals.**

I. Introduction

Support Vector Machines (SVM for short) [1] represent an operational tool widely used by the Machine Learning community. *Per se* a SVM is an n dimensional hyperplane committed to separate positive from negative points of a linearly separable Cartesian space. The success of these machines in comparison with analogous models such as a real-inputs perceptron is due to the algorithm employed to learn them from examples that performs very efficiently and relies on a well defined small subset of examples that it manages in a symbolic way. Thus the algorithm plays the role of a specimen of the computational learning theory [2] allowing theoretical forecasting of the future misclassifying error. This prevision however may result very bad and consequently deprived of any operational consequence. This is because we are generally obliged to broad approximations coming from more or less sophisticated variants of the law of large numbers. In the paper we overcome this drawback working in the Algorithmic Inference framework [3], computing bounds that are linked to the properties of the actual classification instance and typically prove tighter by one order of magnitude in comparison to analogous bounds computed by Vapnik [4]. We numerically check that these bounds delimit slightly oversized confidence intervals [5] for the actual error probability.

The paper is organized as follows: Section II introduces SVMs, while Section III describes and solves the corresponding accuracy estimation problem in the Algorithmic Inference framework. Section IV numerically checks the theoretical results.

II. Learning SVMs

In their basic version, SVMs are used to compute hypotheses in the class H of hyperplanes in $\mathbb{R}^n$, for fixed $n \in \mathbb{N}$. Given a sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \in \mathbb{R}^{mn}$ with associated labels $\{y_1, \ldots, y_m\} \in \{-1, 1\}^m$, the related classification problem lies in finding a *separating hyperplane*, i.e. an $h \in \mathsf{H}$ such that all the points with a given label belong to one of the two half-spaces determined by h.

In order to obtain such a h, an SVM computes first the solution $\{\alpha_1^*, \ldots, \alpha_m^*\}$ of a dual constrained optimization problem

$$\max_{\alpha_1,\ldots,\alpha_m} \quad \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{1}$$

$$\sum_{i=1}^m \alpha_i y_i = 0 \tag{2}$$

$$\alpha_i \geq 0 \quad i = 1, \ldots, m\,, \tag{3}$$

where $\cdot$ denotes the standard dot product in $\mathbb{R}^n$, and then returns a hyperplane (called *separating hyperplane*) whose equation is $\mathbf{w} \cdot \mathbf{x} + b = 0$, where

$$\mathbf{w} = \sum_{i=1}^m \alpha_i^* y_i \mathbf{x}_i \tag{4}$$

$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i \text{ for } i \text{ such that } \alpha_i^* > 0\,. \tag{5}$$

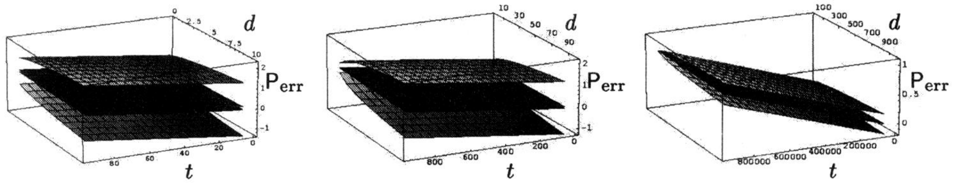
In the case of a *separable sample* (i.e. a sample for which the existence of a separating hyperplane is guaranteed), this algorithm produces a separating hyperplane with *optimal margin*, i.e. a hyperplane maximizing its minimal distance with the sample points. Moreover, typically only a few components of $\{\alpha_1^*, \ldots, \alpha_m^*\}$ are different from zero, so that the hypothesis depends on a small subset of the available examples (those corresponding to non null α's, that are denoted *support vectors* or SV).

A variant of this algorithm, known as *soft-margin classifier* [6], produces hypotheses for which the separability requirement is relaxed, introducing a parameter μ whose value represents an upper bound to the fraction of sample classification errors and a lower bound to fraction of points that are allowed to have a distance less or equal to the margin. The corresponding optimization problem is essentially unchanged, with the sole exception of (3), which now becomes

$$\begin{aligned} 0 \leq \alpha_i \leq \frac{1}{m} \quad i = 1, \ldots, m \\ \sum_{i=1}^m \alpha_i \geq \mu. \end{aligned} \tag{3'}$$

Analogously, the separating hyperplane equation is still obtained through (4-5), though the latter equation needs to be computed on indices i such that $0 < \alpha_i^* < \frac{1}{m}$.

0-7803-9422-4/05/$20.00 ©2005 IEEE



(a) $m = 10^2$ (b) $m = 10^3$ (c) $m = 10^6$

Fig. 1. Comparison between two-sided 0.9 confidence intervals for SVM classification error. t: number of misclassified sample points; d: detail/VC dimension; u: confidence interval extremes; m: sample size. Gray surfaces: VC bounds. Black surfaces: proposed bounds.

III. The associated Algorithmic Inference problem

The theory for computing the bounds lies in a new statistical framework called Algorithmic Inference [3]. The leading idea is to start from specific sample, e.g. a record of observed data constituting the true basis of the available knowledge, and then to infer about the possible continuation of this record – call it population – when the observed phenomenon remains the same. Since we may have many sample suffixes as continuation, we compute their probability distribution on the sole condition of them being *compatible* with the already observed data. This constraint constitutes the inference tool to be molded with the formal knowledge already available on the phenomenon. In particular, in the problem of learning a Boolean function over a space $\mathfrak{X}$, our knowledge base is a *labeled sample* $\mathbf{z}_m = \{(x_i, b_i),\ i = 1, \ldots, m\}$ where m is an integer, $x_i \in \mathfrak{X}$ and b_i are Boolean variables. The formal knowledge stands in a concept class C related to the observed phenomenon, and we assume that for every M and every population $\mathbf{z}_M$ a c exists in C such that $\mathbf{z}_{m+M} = \{(x_i, c(x_i)),\ i = 1, \ldots, m+M\}$. We are interested in another function $\mathscr{A} : \{\mathbf{z}_m\} \mapsto \mathsf{H}$ which, having in input the sample, computes a hypotesis h in a (possibly) different class H such that the random variable $U_{c\div h}$ [1], denoting the measure of the symmetric difference $c \div h$ in the possible continuations of $\mathbf{z}_m$, is low enough with high probability. The bounds we propose in this paper delimit confidence intervals for the mentioned measure, within a dual notion of these intervals where the extremes are fixed and the bounded parameter is a random variable. The core of the theory is the theorem below.

Theorem 3.1: [7] For a space $\mathfrak{X}$ and any probability measure P on it, assume we are given

- concept classes C and H on $\mathfrak{X}$;
- a labeled sample $\mathbf{Z}_m$ drawn from $\mathfrak{X} \times \{0,1\}$;
- a *fairly strongly surjective* function $\mathscr{A} : \{\mathbf{z}_m\} \mapsto \mathsf{H}$

In the case where for any $\mathbf{z}_m$ and any infinite suffix $\mathbf{z}_M$ of it a $c \in \mathsf{C}$ exists computing the example labels of the whole sequence, consider the family of random sets $\{c \in \mathsf{C} : \mathbf{z}_{m+M} = \{(x_i, c(x_i)), i = 1, \ldots, m+M\}$ for any specification $\mathbf{z}_M$ of $\mathbf{Z}_M\}$. Denote $h = \mathscr{A}(\mathbf{z}_m)$ and $U_{c\div h}$ the random variable given by the probability measure of $c \div h$ and $F_{U_{c\div h}}$ its cumulative distribution function. For a given $\mathbf{z}_m$, **if**

- h has *detail* $\mathrm{D}_{(\mathsf{C},\mathsf{H})_h}$ and misclassifies t_h points of $\mathbf{z}_m$,

then for each $\varepsilon \in (0,1)$,

$$\sum_{i=t_h+1}^{m} \binom{m}{i} \varepsilon^i (1-\varepsilon)^{m-i} \geq F_{U_{c\div h}}(\varepsilon) \geq \sum_{i=\mathrm{D}_{(\mathsf{C},\mathsf{H})_h}+t_h}^{m} \binom{m}{i} \varepsilon^i (1-\varepsilon)^{m-i}. \quad (6)$$

Fairly strong surjectivity is a usual regularity condition [3], while $\mathrm{D}_{(\mathsf{C},\mathsf{H})_h}$ is a key parameter of the Algorithmic Inference approach to learning called *detail* [8]. The general idea is that it counts the number of meaningful examples within a sample which prevent $\mathscr{A}$ from computing a hypothesis h' with a wider mistake region $c \div h'$. These points are supposed to be algorithmically computed for each c and h through a *sentry function* $\mathbf{S}$. The maximum $\mathrm{D}_{\mathsf{C},\mathsf{H}}$ of $\mathrm{D}_{(\mathsf{C},\mathsf{H})_h}$ over the entire class $\mathsf{C} \div \mathsf{H}$ of symmetric differences between possible concepts and hypotheses relates to the well known Vapnik-Chervonenkis dimension d_{VC} [4] through the relation: $\mathrm{D}_{\mathsf{C},\mathsf{H}} < \mathrm{d}_{\mathrm{VC}}(\mathsf{C} \div \mathsf{H}) + 1$ [8].

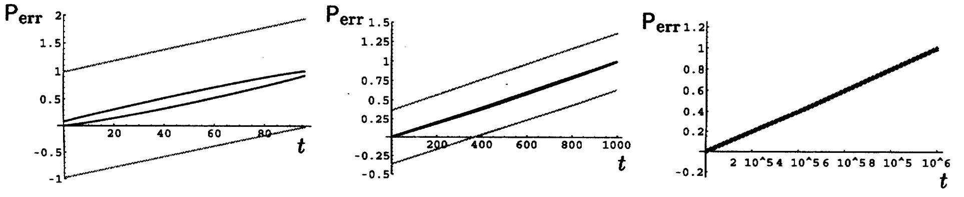
A. The detail of a SVM

The distinctive feature of the hypotheses learnt through SVM is that the detail $\mathrm{D}_{(\mathsf{C},\mathsf{H})_h}$ ranges from 1 to the number μ_h of support vectors minus 1, and its value increases with the broadness of the approximation of the solving algorithm [9].

Lemma 3.2: Let us denote by C the concept class of hyperlanes on a given space $\mathfrak{X}$ and by $\sigma = \{\mathbf{x}_1, \ldots, \mathbf{x}_s\}$ a minimal set of support vectors of a hyperplane h (i.e. σ is a support vector set but, whatever i is, no $\sigma \backslash \{\mathbf{x}_i\}$ does the same). Then, for whatever goal hyperplane c separating the above set accordingly with h, there exists a sentry function $\mathbf{S}$ on $\mathsf{C} \div \mathsf{H}$ and a subset of σ of cardinality at most $s-1$ sentinelling $c \div h$ according to $\mathbf{S}$.

Proof: To identify a hyperplane in an n-dimensional Euclidean space we need to put n *non aligned* points into a linear equations' system, $n+1$ if these points are at a fixed (either negative or positive) distance. This is also the maximum number of support vectors required by a SVM. We may substitute one or more points with direct linear constraints on

[1] By default capital letters (such as U, X) will denote random variables and small letters (u, x) their corresponding specifications; the sets the specifications belong to will be denoted by capital gothic letters $(\mathfrak{U}, \mathfrak{X})$.



(a) $m = 10^2$

(b) $m = 10^3$

(c) $m = 10^6$

Fig. 2. Same comparison as in Fig. 1 with $d = 4$.

the hyperplane coefficients when the topology of the support vectors allows it. Sentinelling the expansion of the symmetric difference $c \div h$ results in forbidding any rotation of h into a h' pivoted along the intersection of c with h. The membership of this intersection to h' adds from 1 to $n-1$ linear relations on its coefficients, so that at most $\#\sigma - 1$ points from σ are necessary (where $\#$ denotes the set cardinality operator), possibly in conjunction with the direct linear constraints on the coefficients to fix h' to h. ■

In synthesis, our approach focuses on a probabilistic description of the *uncertainty region* [10], rather than on its geometric approximation [11].

We must remark that in principle the constraint for h' to contain the intersection of h with c gives rise to $n-1$ linear relations on h' coefficients. These relations may result effective in a shorter number if linear relations occur between them deriving from a linear relation, in own turn, between h and c coefficients. Now, as the former are functions of the sampled points, no way exists for computing coefficients that result exactly in linear relation with those of the unknown (future) c if the sample space is continuous (and its probability distribution does the same). We actually realize these linear relations if either the sample space is discrete or the algorithm computing the hyperplane is so approximate to work on an actually discretised search space. Thus we have the following fact.

Fact 3.1: The number of sentry points of separating hyperplanes computed through support vector machines ranges from 1 to the minimal number of involved support vectors minus one, depending on the approximation with which either sample coordinates are stored or hyperplanes are computed.

B. Confidence intervals for the probability of classifying wrongly

Confidence interval extremes $u_{\text{up}}, u_{\text{dw}}$ for $U_{c \div h}$ with confidence $1 - \delta$ arise for each pair $(\mathrm{D}_{(\mathrm{C},\mathrm{H})_h}, t_h)$ from the solution of the following equations

$$\sum_{i=\mathrm{D}_{(\mathrm{C},\mathrm{H})_h}+t_h}^{m} \binom{m}{i} u_{\text{up}}^i (1-u_{\text{up}})^{m-i} = 1 - \frac{\delta}{2}; \qquad (7)$$

$$\sum_{i=t_h+1}^{m} \binom{m}{i} u_{\text{dw}}^i (1-u_{\text{dw}})^{m-i} = \frac{\delta}{2}. \qquad (8)$$

Figure 1 plots the interval extremes versus detail/VC dimension and number of misclassified points for different sizes of the sample. Companion curves in the figure are the Vapnik bounds [4]:

$$\nu(\mathbf{Z}_m) - 2\sqrt{\frac{d\left(\log\frac{2m}{d}+1\right)-\log\frac{\delta}{9}}{m}} < V(\mathbf{Z}_m) <$$

$$< \nu(\mathbf{Z}_m) + 2\sqrt{\frac{d\left(\log\frac{2m}{d}+1\right)-\log\frac{\delta}{9}}{m}}, \qquad (9)$$

where $V(\mathbf{Z}_m)$ is the random variable measuring $c \div \mathscr{A}(\mathbf{Z}_m)$ according to the same probability (any one) P through which the sample has been drawn, $\nu(\mathbf{Z}_m)$ the corresponding frequency of errors computed from the sample according to $\mathscr{A}$ (*empirical error*), and $d = \mathrm{d}_{\mathrm{VC}}(\mathrm{C} \div \mathrm{H})$. We artificially fill the gap between the two complexity indices by: 1) referring to both complexity indices and empirical error ν constant with concepts and hypotheses, hence $t_h = m\nu$, and 2) assuming $\mathrm{D}_{\mathrm{C},\mathrm{H}} = \mathrm{d}_{VC}(\mathrm{C}) = \mu_h = d$. Analogously, we unify in P_{err} the notations for the error probabilities considered in the two approaches.

The sections of graphs in Fig. 1 with $d = 4$, reported in Fig. 2, show that the two families of bounds asymptotically coincide when m increases, and highlight consistency as a further benefit of our approach, since our intervals are always contained in $[0, 1]$.

IV. Checking the results

The coverage of the above intervals is checked through a huge set of $u_{c \div h}$s sampled from SVM learning instances. The instances are made up of points distributed in the unitary hypercube and labeled according to a series of hyperplanes spanning with a fine discretizing grain all possible hyperecube partitions. In Fig. 3(a) we considered different sample sizes for μ_h fixed to 3. In Fig. 3(b) we conversely maintained the sample size fixed to 100 and considered different μ_h's. Moreover we induced sample classification errors by labeling the sample according to non linear discriminating surfaces (paraboloids). The curves in the figure give the course of the $U_{c \div h}$ bounds with $\mu_h + t_h$. The lower bounds are underrated with a straight line corresponding to $t_h = 0$ in (8). The scattering of experimental points $(\widehat{\mathsf{P}}_{\text{err}}, \log m)$ [2] denotes some overestimation of

[2] $\widehat{\mathsf{P}}_{\text{err}}$ is computed as the frequency of misclassified points from a huge set drawn with the same distribution of the examples.

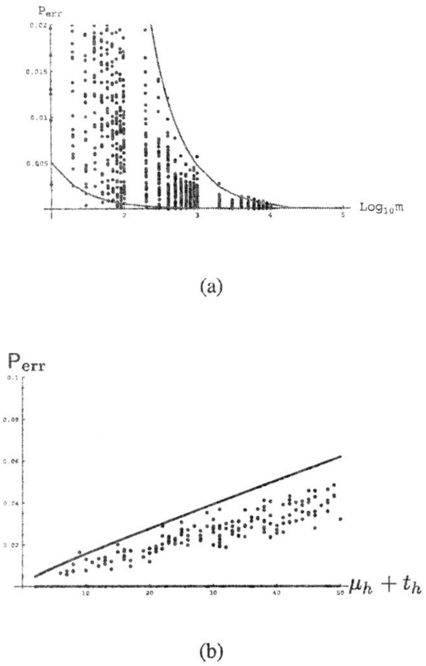


Fig. 3. Course of misclassification probability with the parameters of the learning problem: (a) error probability (P$_{\text{err}}$) vs. sample size, (b) error probability vs. number of support vectors μ_h plus number of misclassifications t_h. Points: sampled values; lines: 0.9 confidence bounds.

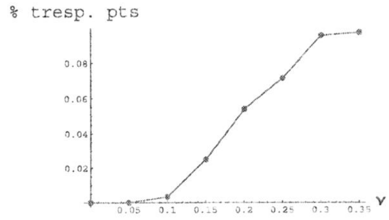


Fig. 4. Course of the experimental confidence level with algorithm approximation.

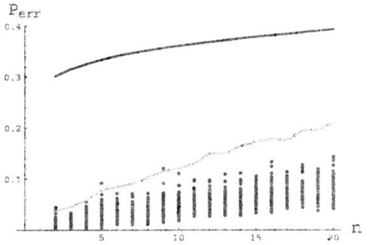


Fig. 5. Course of misclassification probability and related bounds normalized in respect to the dimensionality n of the sample space. Gray line: our bounds; dark line: Rademacher bounds.

the upper bounds. This is due to an analogous overvaluation of $\mathrm{D}_{(\mathsf{C},\mathsf{H})_h}$ through μ_h. As explained before, the gap between the two parameters diminishes with the approximation broadness of the support vector algorithm. This is accounted for in the graph in Fig. 4 which reports the percentage of experiments trespassing the bounds. This quantity comes close to the planned $\delta = 0.1$ with the increase of the free parameter in the *ν-support vector algorithm* [12] employed to learn the hyperplanes. This nicely reflects an expectable rule of thumb according to which true sample complexity of the learning algorithm decreases with the increase of its accuracy.

Bounds based on Rademacher complexity comes from Bartlett inequality [13]

$$\mathrm{P}(Yf(X) \leq 0) \leq \hat{\mathbf{E}}_m \phi(Yf(X)) + \frac{4B}{\gamma m}\sqrt{\sum_{i=1}^{m} k(X_i, X_i)} + \left(\frac{8}{\gamma} + 1\right)\sqrt{\frac{\log(4/\delta)}{2m}} \quad (10)$$

with notation as in the quoted reference, where the first term on the left side corresponds to the empirical margin cost, a quantity that we may assume very close to the empirical error. The computation of this bound requires a sagacious choice of γ and B, which at the best of our preliminary study brought us to the graph in Fig 5. Here we have in abscissas the dimensionality of the instance space. For the sake of comparison, we contrast the above with our upper bounds computed for the same δ (hence coming from a one side confidence interval) and mediated in correspondence with each hypercube dimension, since for each dimension sampled instances may have a different number of support vectors (while t_h we constrained to be 0).

References

[1] C. Cortes and V. Vapnik, "Support-Vector networks," *Machine Learning*, vol. 20, pp. 121–167, 1995.

[2] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 11, no. 27, pp. 1134–1142, 1984.

[3] B. Apolloni, D. Malchiodi, and S. Gaito, *Algorithmic Inference in Machine Learning*. Magill, Adelaide: Advanced Knowledge International, 2003.

[4] V. Vapnik, *Statitical Learning Theory*. New York: John Wiley & Sons, 1998.

[5] S. S. Wilks, *Mathematical Statistics*, ser. Wiley Publications in Statistics. New York: John Wiley, 1962.

[6] B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., *Advances in kernel methods: Support Vector learning*. Cambridge, Mass.: MIT Press, 1999.

[7] B. Apolloni, D. Esposito, Malchiodi, A., C. Orovas, G. Palmas, and J. Taylor, "A general framework for learning rules from data," *IEEE Transactions on Neural Networks*, vol. 15, pp. 1333–1349, 2004.

[8] B. Apolloni and S. Chiaravalli, "PAC learning of concept classes through the boundaries of their items," *Theoretical Computer Science*, vol. 172, pp. 91–120, 1997.

[9] B. Apolloni, S. Bassis, S. Gaito, D. Malchiodi, and A. Minora, "Computing confidence intervals for the risk of a SVM classifier through Algorithmic Inference," in *Biological and Artificial Intelligence Environments*, B. Apolloni, M. Marinaro, and R. Tagliaferri, Eds. Springer, 2005, pp. 225–234.

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

[11] M. Muselli, "Support Vector Machines for uncertainty region detection," in *Neural Nets: WIRN VIETRI '01, 12th Italian Workshop on Neural Nets (Vietri sul Mare, Italy, 17–19 May 2001)*, M. Marinaro and R. Tagliaferri, Eds. London: Springer, 2001, pp. 108–113.

[12] B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett, "New support vector algorithms," *Neural Computations*, vol. 12, pp. 1207–1245, 2000.

[13] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2002.