Margin-Sparsity Trade-Off for the Set Covering Machine

François Laviolette[1], Mario Marchand[1], and Mohak Shah[2]

[1] IFT-GLO, Université Laval,
Sainte-Foy (QC) Canada, G1K-7P4
{Francois.Laviolette, Mario.Marchand}@ift.ulaval.ca
[2] SITE, University of Ottawa,
Ottawa, Ont. Canada, K1N-6N5
mshah@site.uottawa.ca

Abstract. We propose a new learning algorithm for the set covering machine and a tight data-compression risk bound that the learner can use for choosing the appropriate tradeoff between the sparsity of a classifier and the magnitude of its separating margin.

1 Introduction

There exists a wide spectrum of different leaning strategies currently used by learning algorithms to produce classifiers having good generalization. At one end of the spectrum, we have the set covering machine (SCM), proposed by Marchand and Shawe-Taylor (2002), that tries to find the sparsest classifier having few training errors. At the other end of the spectrum, we have the support vector machine (SVM), proposed by Boser et. al. (1992), that tries to find the maximum soft-margin separating hyperplane on the training data. Since both of these learning machines can produce classifiers having good generalization, it is worthwhile to investigate if classifiers with improved generalization could be found by learning algorithms that try to optimize a non-trivial function that depends on both the sparsity of a classifier and the magnitude of its separating margin.

There seems to be a widespread belief that learning algorithms should somehow try to find such a non-trivial margin-sparsity trade-off. For example, to find a sparser SVM (but with a smaller margin), Bennett (1999) and Bi et. al. (2003) have proposed to minimize an ℓ_1-norm functional (instead of the traditional ℓ_2-norm) and have found that, indeed, the sparser SVM sometimes had better generalization. Therefore, from this SVM perspective, we should consider algorithms that minimizes an ℓ_β-norm functional for any $\beta \in [0, 2]$. In the $\beta = 2$ limit, we obtain the SVM with the largest possible separating margin (without considering its sparsity). In the $\beta = 0$ limit, we would obtain the sparsest SVM (without considering the magnitude of its separating margin). This parameter β would then control the margin-sparsity trade-off of the final classifier. Unfortunately, this optimization problem is currently efficiently solvable only for $\beta = 2$ and 1.

This computational difficulty does not arise (so abruptly) if, instead, we consider margin-sparsity trade-off learning algorithms from the SCM perspective. Indeed, the learning algorithm for the SCM proposed by Marchand and Shawe-Taylor (2002) consists of a set covering greedy heuristic that, at each greedy step, appends, to a conjunction, the Boolean-valued feature that covers the largest number of negative examples

J. Gama et al. (Eds.): ECML 2005, LNAI 3720, pp. 206–217, 2005.

without making too many errors on the positive examples. If, in addition, we force the algorithm to use only features having the property that all the (remaining) training examples are at least a distance γ from its decision surface, we are assured that a conjunction of such features will give a classifier having no training examples within a distance γ of its decision surface. In the $\gamma = 0$ limit, the goal of the learner is to produce the sparsest SCM without considering the magnitude of its separating margin (as in the original SCM algorithm). For finite γ, we will achieve a separating margin of at least γ at the expense of having more features in the SCM. Hence, γ is a parameter that controls the margin-sparsity trade-off of the final classifier without introducing any substantial computational difficulty. We therefore propose, in Section 3, a margin-sparsity trade-off learning algorithm for the SCM which was inspired by this simple idea.

The widespread belief that learning algorithms should try to find a non-trivial margin-sparsity trade-off is, to our knowledge, not currently supported by a generalization error bound (also called risk bound) that explicitly depends on *both* the sparsity of a classifier *and* the magnitude of its separating margin. However, both sparsity and margin can be considered as different forms of data-compression. Indeed, sparsity is a form of data compression known as sample-compression (Littlestone and Warmuth, 1986) since it means that a classifier can be reconstructed from a small subset of the training data. Less obviously, the magnitude of the separating margin of a classifier can also be considered as a form of data compression since it means that there exists a small code that can specify a "good" location for the classifier's decision surface. For the SCM of Marchand and Shawe-Taylor (2002), each *data-dependent ball* feature is identified by two training points: a *center* and a *border* (to define the radius of the ball). In section 3, we propose instead to code the radius of each ball by a *message string*. Hence, the existence of a large margin of "equally good radius values" for a ball will imply the existence of a short code for its radius. With this new version of the SCM, we therefore identify each classifier by two distinct information sources: a *compression set* which consists of the center of each ball in the classifier and a *message string* which encodes the radius value of each ball.

In section 2 of this paper, we therefore propose a tight data-compression risk bound that depends explicitly on these two information sources. This bound therefore exhibits a non trivial trade-off between sparsity (the inverse of the compression set size) and the margin (the inverse of the message length) that classifiers should attempt to optimize on the training data. In contrast with other sample-compression bounds, the proposed bound is valid for any compression set-dependent distribution of messages and, as we argue, permits the usage of smaller message strings which, in turn, can help reduce significantly the size of the risk bound. We then show, in section 3, how we can apply this risk bound to the SCM by providing an appropriate compression set-dependent distribution of messages. Finally, we show, on natural data sets, that the new SCM algorithm compares favorably to the SCM algorithm of Marchand and Shawe-Taylor (2002) and we also show that the data-compression risk bound is an effective guide for choosing the proper margin-sparsity trade-off of a classifier.

2 A Data-Compression Risk Bound

We consider binary classification problems where the input space $\mathcal{X}$ consists of an arbitrary subset of $\mathbb{R}^n$ and the output space $\mathcal{Y} = \{-1, +1\}$. An example $\mathbf{z} \stackrel{\text{def}}{=} (\mathbf{x}, y)$ is an input-output pair where $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$. We are interested in learning algorithms that have the following property. Given a training set $S = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ of m examples, the classifier $A(S)$ returned by algorithm A is described entirely by two *complementary sources of information*: a subset $\mathbf{z_i}$ of S, called the *compression set*, and a *message string* σ which represents the additional information needed to obtain a classifier from the compression set $\mathbf{z_i}$.

Given a training set S, the compression set $\mathbf{z_i}$ is defined by a vector $\mathbf{i}$ of indices $\mathbf{i} \stackrel{\text{def}}{=} (i_1, i_2, \ldots, i_{|\mathbf{i}|})$ with $i_j \in \{1, \ldots, m\}\ \forall j$ and $i_1 < i_2 < \ldots < i_{|\mathbf{i}|}$ and where $|\mathbf{i}|$ denotes the number of indices present in $\mathbf{i}$. Hence, $\mathbf{z}_i$ denotes the ith example of S whereas $\mathbf{z_i}$ denotes the subset of examples of S that are pointed by the vector of indices $\mathbf{i}$ defined above. We will use $\overline{\mathbf{i}}$ to denote the set of indices not present in $\mathbf{i}$. Hence, we have $S = \mathbf{z_i} \cup \mathbf{z_{\overline{i}}}$ for any vector $\mathbf{i} \in \mathcal{I}$ where $\mathcal{I}$ denotes the set of the 2^m possible realizations of $\mathbf{i}$.

The fact that any classifier returned by algorithm A is described by a compression set and a message string implies that there exists a *reconstruction function* $\mathcal{R}$, associated with A, that outputs a classifier $\mathcal{R}(\sigma, \mathbf{z_i})$ when given an arbitrary compression set $\mathbf{z_i} \subseteq S$ and message string σ chosen from the set $\mathcal{M}(\mathbf{z_i})$ of all distinct messages that can be supplied to $\mathcal{R}$ with the compression set $\mathbf{z_i}$. It is only when such a $\mathcal{R}$ exists that the classifier returned by $A(S)$ is *always* identified by a compression set $\mathbf{z_i}$ and a message string σ.

The perceptron learning rule and the SVM are examples of learning algorithms where the final classifier can be reconstructed solely from a compression set (Graepel. et. al.,2000, 2001). In contrast, the reconstruction function for SCMs needs both a compression set and a message string. Later, we will see how the learner can trade-off the compression set size with the length of the message string to obtain a classifier with a smaller risk bound and, hopefully, a smaller true risk.

We seek a tight risk bound for arbitrary reconstruction functions that holds uniformly for all compression sets and message strings. For this, we adopt the PAC setting where each example $\mathbf{z}$ is drawn according to a fixed, but unknown, probability distribution D on $\mathcal{X} \times \mathcal{Y}$. The risk $R(f)$ of any classifier f is defined as the probability that it misclassifies an example drawn according to D:

$$R(f) \stackrel{\text{def}}{=} \Pr_{(\mathbf{x},y)\sim D}\left(f(\mathbf{x}) \neq y\right) = \mathbf{E}_{(\mathbf{x},y)\sim D} I(f(\mathbf{x}) \neq y)$$

where $I(a) = 1$ if predicate a is true and 0 otherwise. Given a training set $S = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ of m examples, the *empirical risk* $R_S(f)$ on S, of any classifier f, is defined according to:

$$R_S(f) \stackrel{\text{def}}{=} \frac{1}{m}\sum_{i=1}^{m} I(f(\mathbf{x}_i) \neq y_i) \stackrel{\text{def}}{=} \mathbf{E}_{(\mathbf{x},y)\sim S} I(f(\mathbf{x}) \neq y)$$

Let $\mathbf{Z}^m$ denote the collection of m random variables whose instantiation gives a training sample $S = \mathbf{z}^m = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$. Let us denote $\Pr_{\mathbf{Z}^m \sim D^m}(\cdot)$ by $\mathbf{P}_{\mathbf{Z}^m}(\cdot)$. To obtain

the tightest possible risk bound, we fully exploit the fact that the distribution of classification errors is a binomial. The binomial tail distribution $\mathrm{Bin}(\frac{k}{m}, r)$ associated with a classifier of (true) risk r is defined as the probability that this classifier makes at most k errors on a test set of m examples: $\mathrm{Bin}\left(\frac{k}{m}, r\right) \stackrel{\mathrm{def}}{=} \sum_{i=0}^{k} \binom{m}{i} r^i (1-r)^{m-i}$.

Following Langford (2005) and Blum and Langford (2003), we now define the *binomial tail inversion* $\overline{\mathrm{Bin}}\left(\frac{k}{m}, \delta\right)$ as the largest risk value that a classifier can have while still having a probability of at least δ of observing at most k errors out of m examples:

$$\overline{\mathrm{Bin}}\left(\frac{k}{m}, \delta\right) \stackrel{\mathrm{def}}{=} \sup\left\{r : \mathrm{Bin}\left(\frac{k}{m}, r\right) \geq \delta\right\}$$

From this definition, it follows that $\overline{\mathrm{Bin}}\left(R_S(f), \delta\right)$ is the *smallest* upper bound, which holds with probability at least $1-\delta$, on the true risk of any classifier f with an observed empirical risk $R_S(f)$ on a test set of m examples:

$$\mathbf{P}_{\mathbf{Z}^m}\left\{R(f) \leq \overline{\mathrm{Bin}}\Big(R_{\mathbf{Z}^m}(f), \delta\Big)\right\} \geq 1-\delta \quad \forall f \tag{1}$$

Note that the quantifier $\forall f$ appears *outside* the probability $\mathbf{P}_{\mathbf{Z}^m}\{\cdot\}$ because the bound $\overline{\mathrm{Bin}}\left(R_S(f), \delta\right)$ does not hold *simultaneously* (and uniformly) for all classifiers f member of some predefined class $\mathcal{F}$. In contrast, the proposed risk bound of Theorem 1 holds uniformly for all compression sets and message strings.

The proposed risk bound is a generalization of the sample-compression risk bound of Langford (2005) to the case where part of the data-compression information is given by a message string. It also has the property to reduce to the Occam's razor bound when the compression set $\mathbf{z_i}$ vanishes. The idea of using a message string as an additional source of information was also used by Littlestone and Warmuth (1986) and Ben-David and Litman (1998) to obtain a sample-compression bound looser than the bound presented here. Moreover, in contrast with these bounds, Theorem 1 applies to any compression set-dependent distribution of messages $P_{\mathcal{M}(\mathbf{z_i})}$ satisfying:

$$\sum_{\sigma \in \mathcal{M}(\mathbf{z_i})} P_{\mathcal{M}(\mathbf{z_i})}(\sigma) \leq 1 \quad \forall \mathbf{z_i} \tag{2}$$

and any prior distribution $P_{\mathcal{I}}$ of vectors of indices satisfying:

$$\sum_{\mathbf{i} \in \mathcal{I}} P_{\mathcal{I}}(\mathbf{i}) \leq 1 \tag{3}$$

Theorem 1. *For any reconstruction function $\mathcal{R}$ that maps arbitrary subsets of a training set and message strings to classifiers, for any prior distribution $P_{\mathcal{I}}$ of vectors of indices, for any compression set-dependent distribution of messages $P_{\mathcal{M}(\mathbf{z_i})}$, and for any $\delta \in (0, 1]$, we have:*

$$\mathbf{P}_{\mathbf{Z}^m}\Bigg\{\forall \mathbf{i} \in \mathcal{I}, \forall \sigma \in \mathcal{M}(\mathbf{Z_i})\colon R(\mathcal{R}(\sigma, \mathbf{Z_i})) \leq \overline{\mathrm{Bin}}\Big(R_{\mathbf{Z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z_i})), P_{\mathcal{I}}(\mathbf{i}) P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta\Big)\Bigg\} \geq 1-\delta$$

where, for any training set $\mathbf{z}^m$, $R_{\mathbf{z}_{\bar{\mathbf{i}}}}(f)$ *denotes the empirical risk of classifier* f *on the examples of* $\mathbf{z}^m$ *that do not belong to the compression set* $\mathbf{z}_{\mathbf{i}}$.

Proof. Consider:

$$P' \stackrel{\text{def}}{=} \mathbf{P}_{\mathbf{Z}^m}\left\{ \exists \mathbf{i} \in \mathcal{I} \colon \exists \sigma \in \mathcal{M}(\mathbf{Z}_{\mathbf{i}}) \colon R(\mathcal{R}(\sigma, \mathbf{Z}_{\mathbf{i}})) > \overline{\mathrm{Bin}}\Big(R_{\mathbf{Z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z}_{\mathbf{i}})), P_{\mathcal{I}}(\mathbf{i}) P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}(\sigma)\delta\Big) \right\}$$

To prove the theorem, we show that $P' \leq \delta$. Since $\mathbf{P}_{\mathbf{Z}^m}(\cdot) = \mathbf{E}_{\mathbf{Z}_{\mathbf{i}}} \mathbf{P}_{\mathbf{Z}_{\bar{\mathbf{i}}}|\mathbf{Z}_{\mathbf{i}}}(\cdot)$, the union bound and Equations 1, 2, and 3 imply that we have:

$$P' \leq \sum_{\mathbf{i}\in\mathcal{I}} \mathbf{E}_{\mathbf{Z}_{\mathbf{i}}} \sum_{\sigma\in\mathcal{M}(\mathbf{Z}_{\mathbf{i}})} \mathbf{P}_{\mathbf{Z}_{\bar{\mathbf{i}}}|\mathbf{Z}_{\mathbf{i}}} \left\{ R(\mathcal{R}(\sigma, \mathbf{Z}_{\mathbf{i}})) > \overline{\mathrm{Bin}}\Big(R_{\mathbf{Z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z}_{\mathbf{i}})), P_{\mathcal{I}}(\mathbf{i}) P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}(\sigma)\delta\Big) \right\}$$

$$\leq \sum_{\mathbf{i}\in\mathcal{I}} \mathbf{E}_{\mathbf{Z}_{\mathbf{i}}} \sum_{\sigma\in\mathcal{M}(\mathbf{Z}_{\mathbf{i}})} P_{\mathcal{I}}(\mathbf{i}) P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}(\sigma)\delta \;\leq\; \delta \qquad \blacksquare$$

The risk bound of Theorem 1 appears to be as tight as it possibly can. Indeed, the proof of Theorem 1 contains three inequalities. The last two inequalities come from Equations 1, 2, and 3 and cannot be improved. The first inequality comes from the application of the union bound for all the possible choices of a compression subset of the training set and is unavoidable for statistically independent training examples.

It is important to note that, once $P_{\mathcal{I}}$ and $P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}$ are specified, the risk bound of Theorem 1 for classifier $\mathcal{R}(\mathbf{z}_{\mathbf{i}}, \sigma)$ depends on its empirical risk *and* on the product $P_{\mathcal{I}}(\mathbf{i}) P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}(\sigma)$. However, $\ln\left(\frac{1}{P_{\mathcal{I}}(\mathbf{i}) P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}(\sigma)}\right)$ is just the amount of information needed to specify a classifier $\mathcal{R}(\mathbf{z}_{\mathbf{i}}, \sigma)$ once we are given a training set and the priors $P_{\mathcal{I}}$ and $P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}$. The $\ln(1/P_{\mathcal{I}}(\mathbf{i}))$ term is the information content of the vector of indices $\mathbf{i}$ that specifies the compression set and the $\ln(1/P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}(\sigma))$ term is the information content of the message string σ. Consequently the bound of Theorem 1 specifies quantitatively how much training errors learning algorithms should trade-off with the amount of information needed to specify a classifier by $\mathbf{i}$ and σ.

Any bound expressed in terms of the binomial tail inversion can be turned into a more conventional and looser bound by inverting a standard approximation of the binomial tail such as those obtained from the inequalities of Chernoff and Hoeffding. In this paper, we make use of the following approximations (provided here without proof) for the binomial tail inversion:

Lemma 1. *For any integer* $m \geq 1$ *and* $k \in \{0, \ldots, m\}$, *we have:*

$$\overline{\mathrm{Bin}}\left(\frac{k}{m}, \delta\right) \leq 1 - \exp\left(\frac{-1}{m-k}\left[\ln\binom{m}{k} + \ln\left(\frac{1}{\delta}\right)\right]\right) \tag{4}$$

$$\leq \frac{1}{m-k}\left[\ln\binom{m}{k} + \ln\left(\frac{1}{\delta}\right)\right] \tag{5}$$

Therefore, these approximations enable us to rewrite the bound of Theorem 1 into the following looser (but somewhat clearer and more conventional) form:

Corollary 1. *For any reconstruction function $\mathcal{R}$ that maps arbitrary subsets of a training set and message strings to classifiers, for any prior distribution $P_{\mathcal{I}}$ of vectors of indices, for any compression set-dependent distribution of messages $P_{\mathcal{M}(\mathbf{z_i})}$, and for any $\delta \in (0, 1]$, we have:*

$$\mathbf{P}_{\mathbf{Z}^m}\left\{\forall \mathbf{i} \in \mathcal{I}, \forall \sigma \in \mathcal{M}(\mathbf{Z_i})\colon R(\mathcal{R}(\sigma, \mathbf{Z_i})) \leq 1 - \exp\left(\frac{-1}{m-d-k}\left[\ln\binom{m-d}{k} + \ln\left(\frac{1}{P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta}\right)\right]\right)\right\} \geq 1-\delta \quad (6)$$

and, consequently:

$$\mathbf{P}_{\mathbf{Z}^m}\left\{\forall \mathbf{i} \in \mathcal{I}, \forall \sigma \in \mathcal{M}(\mathbf{Z_i})\colon R(\mathcal{R}(\sigma, \mathbf{Z_i})) \leq \frac{1}{m-d-k}\left[\ln\binom{m-d}{k} + \ln\left(\frac{1}{P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta}\right)\right]\right\} \geq 1-\delta \quad (7)$$

where $d \stackrel{\text{def}}{=} |\mathbf{i}|$ is the sample compression set size of classifier $\mathcal{R}(\sigma, \mathbf{Z_i})$ and $k \stackrel{\text{def}}{=} |\bar{\mathbf{i}}|R_{\mathbf{z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z_i}))$ is the number of training errors that this classifier makes on the examples that are not in the compression set.

It is now quite clear from Corollary 1 that the risk bound of classifier $\mathcal{R}(\sigma, \mathbf{Z_i})$ is small when its compression set size d and its number k of training errors are both much smaller than the number m of training examples. These are uniform bounds over a set of data-dependent classifiers defined by the reconstruction function $\mathcal{R}$. In contrast, VC bounds (Vapnik 1998) and Rademacher bounds (Mendelson, 2002) are uniform bounds over a set of functions defined *without reference to the training data*. Hence, these latter bounds do not apply to our case.

The bound of Equation 6 is very similar to (and slightly tighter than) the recent bound of Marchand and Sokolova (2005).

The looser bound of Equation 7 is similar to the bounds of Littlestone and Warmuth (1986) and Floyd and Warmuth (1995) when the set $\mathcal{M}$ of all possible messages is independent of the compression set $\mathbf{z_i}$ and when we choose:

$$P_{\mathcal{M}(\mathbf{z_i})}(\sigma) = 1/|\mathcal{M}| \quad \forall \sigma \in \mathcal{M} \quad (8)$$

$$P_{\mathcal{I}}(\mathbf{i}) = \binom{m}{|\mathbf{i}|}^{-1}(m+1)^{-1} \quad \forall \mathbf{i} \in \mathcal{I} \quad (9)$$

But other choices that give better bounds are clearly possible. For example, in the following sections we will use:

$$P_{\mathcal{I}}(\mathbf{i}) = \binom{m}{|\mathbf{i}|}^{-1}\zeta(|\mathbf{i}|) \quad \text{with} \quad \zeta(a) \stackrel{\text{def}}{=} \frac{6}{\pi^2}(a+1)^{-2} \quad \forall a \in \mathbb{N} \quad (10)$$

which satisfies the constraint of Equation 3 since $\sum_{i=1}^{\infty} i^{-2} = \pi^2/6$. This choice for $P_{\mathcal{I}}$ has the advantage that the risk bounds do not deteriorate too rapidly when $|\mathbf{i}|$ increases.

In the next section, we show how we can apply the risk bounds of Theorem 1 and Corollary 1 to the SCM. For this task, we will provide choices for the distribution of messages $P_{\mathcal{M}(\mathbf{z_i})}$ which are more appropriate than the simplest choice given by Equation 8. Indeed, we feel that it is important to allow the set of messages to depend on the sample compression $\mathbf{z_i}$ since it is conceivable that for some $\mathbf{z_i}$, very little extra information may be needed to identify the classifier whereas for some other $\mathbf{z_i}$, more information may be needed. Without such a dependency on $\mathbf{z_i}$, the set of possible messages $\mathcal{M}$ would be unnecessarily large and would loosen the risk bound. But, more importantly, the risk bound would not depend on the particular message σ used. However, we feel that it is important for learning algorithms to be able to trade-off the complexity (or information content) of $\mathbf{i}$ with the complexity of σ. Hence, a good risk bound should somehow indicate what the proper trade-off should be.

3 Application to the Set Covering Machine

Recall that the task of the SCM (Marchand and Shawe-Taylor 2002) is to construct the smallest possible conjunction of (Boolean-valued) features. We discuss here only the conjunction case. The disjunction case is treated similarly just by exchanging the role of the positive with the negative examples.

For the case of *data-dependent balls*, each feature is identified by a training example, called a *center* $(\mathbf{x}_c, y_c)$, and a radius ρ. Given any metric d, the output $h(\mathbf{x})$ on any input example $\mathbf{x}$ of such a feature is given by:

$$h(\mathbf{x}) = \begin{cases} y_c & \text{if } d(\mathbf{x}, \mathbf{x}_c) \leq \rho \\ -y_c & \text{otherwise} \end{cases}$$

3.1 Coding Each Radius with a Training Example

Marchand and Shawe-Taylor (2002) have proposed to use another training example $\mathbf{x}_b$, called a *border point*, to code for the radius so that $\rho = d(\mathbf{x}_c, \mathbf{x}_b)$. In this case, given a compression set $\mathbf{z_i}$, we need to specify the examples in $\mathbf{z_i}$ that are used for a border point without being used as a center. As explained by Marchand and Shawe-Taylor (2002), no additional amount of information is required to pair each center with its border point whenever the reconstruction function $\mathcal{R}$ is constrained to produce a classifier that always correctly classifies the compression set. Furthermore, as argued by Marchand and Shawe-Taylor (2002), we can limit ourselves to the case where each border point is a positive example. In that case, each message $\sigma \in \mathcal{M}(\mathbf{z_i})$ just needs to specify the positive examples that are a border point without being a center. Let $n(\mathbf{z_i})$ and $p(\mathbf{z_i})$ be, respectively, the number of negative and the number of positive examples in compression set $\mathbf{z_i}$. Let $b(\sigma)$ be the number of border point examples specified in message σ and let $\zeta(a)$ be the same as defined in Equation 10. We can then use:

$$P_{\mathcal{M}(\mathbf{z_i})}(\sigma) = \zeta(b(\sigma)) \cdot \binom{p(\mathbf{z_i})}{b(\sigma)}^{-1} \tag{11}$$

since, in that case, we have for any compression set $\mathbf{z_i}$:

$$\sum_{\sigma \in \mathcal{M}(\mathbf{z_i})} P_{\mathcal{M}(\mathbf{z_i})}(\sigma) = \sum_{b=0}^{p(\mathbf{z_i})} \zeta(b) \sum_{\sigma : b(\sigma)=b} \binom{p(\mathbf{z_i})}{b(\sigma)}^{-1} \leq 1$$

With this distribution $P_{\mathcal{M}(\mathbf{z_i})}$, the risk bound of Theorem 1 is tighter than the one provided by Marchand and Shawe-Taylor (2002) because of the more efficient treatment of the training errors made by using the binomial tail inversion.

3.2 Coding Each Radius with a Small Message String

Another alternative, not considered by Marchand and Shawe-Taylor (2002), is to code each radius value by a message string having the fewest number of bits. In this case, no border points are used and the compression set only consists of ball centers. Consequently, the risk bounds of Theorem 1 and Corollary 1 will be smaller for classifiers described by this method provided that we do not use to many bits to code each radius. We expect that this will be the case whenever there exists a large interval $[r_1, r_2]$ (*i.e.*, a margin) of radius values such that no training examples are present between the two concentric spheres, centered on $\mathbf{x}_c$, with radius r_1 and r_2. The best radius value in that case will be the one that has the shortest code. A similar idea was applied by von Luxburg et. al. (2004) for coding the maximum-margin hyperplane solution for support vector machines.

Hence, consider the problem of coding a radius value $r \in [r_1, r_2] \subset [0, R]$ where R is some predefined value that cannot be exceeded and where $[r_1, r_2]$ is an interval of "equally good" radius values[1]. We propose the following diadic coding scheme for the identification of a radius value that belongs to that interval. Let l be the number of bits that we use for the code. We adopt the convention that a code of $l = 0$ bits specifies the radius value $R/2$. A code of $l = 1$ bit either specifies the value $R/4$ (when the bit is 0) or the value $3R/4$ (when the bit is 1). A code of $l = 2$ specifies one of the following values: $R/8, 3R/8, 5R/8, 7R/8$. Hence, a code of l bits specifies one value among the set Λ_l of radius values:

$$\Lambda_l \stackrel{\text{def}}{=} \left\{ \frac{2j-1}{2^{l+1}} R \right\}_{j=1}^{2^l}$$

Given an interval $[r_1, r_2] \subset [0, R]$ of radius values, we take the smallest number l of bits such that there exists a radius value in Λ_l that falls in the interval $[r_1, r_2]$. In this way, we will need at most $\lfloor \log_2(R/(r_2 - r_1)) \rfloor$ bits to obtain a radius value that falls in $[r_1, r_2]$.

Hence, to specify the radius for each center of a compression set, we need to specify the number l of bits and a l-bit string s that identifies one of the radius values in Λ_l. Therefore, the message string σ sent to the reconstruction function $\mathcal{R}$, for a compression set $\mathbf{z_i}$, consists of the set of pairs (l_i, s_i) of numbers needed to identify the radius of each center $i \in \mathbf{i}$. The risk bound does not depend on how we actually code σ

[1] By a "good" radius value, we mean a radius value for a ball that would cover many negative examples and very few positive examples (see the learning algorithm).

(for some receiver). It only depends on the a priori probabilities assigned to each possible realization of σ. We choose the following distribution:

$$P_{\mathcal{M}(\mathbf{Z_i})}(\sigma) \stackrel{\text{def}}{=} P_{\mathcal{M}(\mathbf{Z_i})}(l_1, s_1, \ldots, l_{|\mathbf{i}|}, s_{|\mathbf{i}|}) = \prod_{i \in \mathbf{i}} \zeta(l_i) \cdot 2^{-l_i} \tag{12}$$

where $\zeta(l_i)$ is the same as given in Equation 10.

Note that by giving equal a priori probability to each of the 2^{l_i} strings s_i of length l_i, we give no preference to any radius value in Λ_{l_i} once we have chosen a scale R that we believe is appropriate. The distribution ζ that we have chosen for each string length l_i has the advantage of decreasing slowly so that the risk bound does not deteriorate to rapidly as l_i increases. Other choices are clearly possible.

By comparing the risk bounds of Corollary 1 for the two possible choices we have for coding each radius (either with an example or with a message string), we notice that it should be preferable to code explicitly a radius value with a string whenever we use a number l of bits less than $\log_2 m$ (roughly). Hence, this will be the case whenever there exists an interval $[r_1, r_2]$ of "good" radius values such that $(r_2 - r_1)/R \gtrapprox 1/m$.

Finally, we emphasize that the risk bounds of Theorem 1 and Corollary 1, used in conjunction with the distribution of messages given by Equation 12, provides a guide for choosing the appropriate trade-off between sparsity (the inverse of the size of the compression set) and margin (the inverse of the length of the message string). Indeed, the risk bound for an SCM with a decision surface having a large margin of separation (small l_is) may be smaller than the risk bound of a sparser SCM having a smaller margin (large l_is).

4 The Learning Algorithm

Ideally, we would like to find a conjunction of balls that minimizes the risk bound of Theorem 1 with the distribution given by Equation 12. Unfortunately, this cannot be done efficiently in all cases since this problem is at least as hard as the (NP-complete) minimum set cover problem (Marchand and Shawe-Taylor 2002). However, the simple *set covering greedy heuristic* will construct a conjunction of at most $r \ln(m)$ balls whenever there exists a conjunction of r balls that makes no errors with a training set of m examples (Marchand and Shawe-Taylor 2002).

We say that a ball *covers* an example iff it assigns `-1` to that example. The set covering greedy heuristic simply consists of using a ball that covers the largest number of negative examples (without making any errors on the positives), remove these negative covered examples and repeat until all the negative examples are covered. Marchand and Shawe-Taylor (2002) have modified this heuristic by incorporating the possibility of making training errors if the final classifier is much smaller. It can be described as follows. Let N be the set of negative examples and P be the set of positive examples. We start with $N' = N$ and $P' = P$. Let Q_i be the subset of N' covered by ball i and let R_i be the subset of P' covered by ball i. We choose the ball i that maximizes the *utility* U_i defined as:

$$U_i \stackrel{\text{def}}{=} |Q_i| - p \cdot |R_i| \tag{13}$$

where p is the *penalty* suffered by covering (and hence, misclassifying) a positive example. Once we have found a ball maximizing U_i, we update $N' = N' - Q_i$ and $P' = P' - R_i$ and repeat to find the next ball until either $N' = \emptyset$ or the maximum number v of balls has been reached (early stopping the greedy).

Here we first modify the heuristic of Marchand and Shawe-Taylor (2002) by allowing a maximum number of bits l^* that can be used for coding the radius of each ball. Classifiers obtained with a small value of l^* will, on average, have a large separating margin. Moreover, for this new learning algorithm, the distribution of messages given by Equation 12 is defined for a fixed value of R (the "predefined radius value that cannot be exceeded"). Hence, in this case, R should be chosen from the *definition* of each input attribute *without observing the data*. Consequently, this will generally force *each ball* of the classifier to use a large number of bits for its radius value; otherwise the final classifier is likely to make numerous training errors. We have therefore used the following scheme to choose R *from the training data*. We first choose a value R^* from the definition of each input attribute (without observing the data). This could be $R^* = \sqrt{n}$ for the case of n $\{0,1\}$-valued attributes. Then, we consider t equally-spaced values for R in the interval $]0, R^*]$. The message string σ described in Section 3.2 is then just preceded by the index to one of these t possible values. The value of R referred to by this index will then be used for *every ball* of the classifier. For this extra part of the message, we have assigned equal probability to each of the t possible values for R. With this scheme, we only need to multiply $P_{\mathcal{M}(\mathbf{z_i})}(\sigma)$ of Equation 12 by $1/t$. Nevertheless, this introduces one more adjustable parameter in the learning algorithm: the value of R.[2] Therefore, p, v, l^*, and R are the "learning parameters" that our heuristic uses to generate a set of classifiers. At the end, we can use the bound of Theorem 1 to select the best classifier. Another alternative is to determine the best parameter values by cross-validation.

5 Empirical Results on Natural Data

We have compared the new learning algorithm (called here SCM2), that codes each ball radius with a message string, with the old algorithm (called here SCM1), that codes each radius with a training example. Both of these algorithms were also compared with the support vector machine (SVM) equipped with a RBF kernel of variance $1/2\gamma$ and a soft margin parameter C. Each SCM algorithm used the L_2 metric since this is the metric present in the argument of the RBF kernel.

Each algorithm was tested on the UCI data sets of Table 1. Each data set was randomly split in two parts. About half of the examples was used for training and the remaining set of examples was used for testing. The corresponding values for these numbers of examples are given in the "train" and "test" columns of Table 1. The learning parameters of all algorithms were determined from the training set *only*. The parameters C and γ for the SVM were determined by the 5-fold cross validation (CV) method performed on the training set. The parameters that gave the smallest 5-fold CV error were then used to train the SVM on the whole training set and the resulting classifier was then run on the testing set. Exactly the same method (with the same 5-fold

[2] We have used $t \approxeq 30$ different values of R in our experiments.

Table 1. SVM and SCM results on UCI data sets

Data Set			SVM results				SCM1-cv		SCM1-b		SCM2-cv			SCM2-b		
Name	train	test	C	γ	SVs	errs	b	errs	b	errs	b	l^*	errs	b	l^*	errs
breastw	343	340	1	0.1	38	15	2	11	1	12	1	3	12	1	1	12
bupa	170	175	2	3.0	169	66	2	71	2	70	2	7	69	11	7	67
credit	353	300	100	0.25	282	51	12	65	1	57	11	6	49	8	5	46
haberman	144	150	2	0.5	81	39	2	41	1	39	8	2	36	2	2	37
pima	400	368	0.5	0.02	241	96	1	108	1	105	4	1	107	13	5	103
USvotes	235	200	1	0.02	53	13	8	26	3	19	7	3	19	4	2	15
Hart	150	147	1	3.0	64	26	1	28	1	23	1	2	24	1	2	23
Glass	107	107	10	3.0	51	29	4	20	4	19	7	6	19	3	5	18

split) was used to determine the learning parameters of both SCM1 and SCM2. These results are referred to (in Table 1) as SCM1-cv and SCM2-cv. In addition to this, we have compared this 5-fold CV model selection method with a model selection method that uses the risk bound 6 of Corollary 1 to select the best SCM classifier obtained from the *same* possible choices of the learning parameters that we have used for the 5-fold CV method. The SCM that minimizes the risk bound (computed from the training set) was then run on the testing set. These results are referred to (in Table 1) as SCM1-b and SCM2-b. For SCM1, the risk bound was used in conjunction with the distribution of messages given by Equation 11. For SCM2, the risk bound was used in conjunction with the distribution of messages given by Equation 12.

The SVM results are reported in Table 1 where the "SVs" column refers to the number of support vectors present in the final classifier and the "errs" column refers to the number of classification errors obtained on the testing set. This last notation is used also for all the SCM results reported in Table 1. In addition to this, the "b" and "l^*" columns refer, respectively, to the number of balls and the maximum number of bits used by the final classifier.

We observe that SCMs are always much sparser than SVMs with roughly the same generalization error. Moreover, the risk bound is often better than 5-fold CV for choosing the classifier with the smallest generalization error. (We have observed that the risk bound was almost always within a factor of three of the test error.) We also observe that SCM2 is generally as good as, and sometimes clearly better than, SCM1 for producing classifiers with a small generalization error. Finally, it is interesting to note the strong tendency of SCM2 to produce classifiers with more balls than those produced by SCM1. This is especially true for SCM2-b versus SCM1-b. Hence SCM2 generally sacrifices sparsity to obtain a larger margin.

6 Conclusion

We have proposed a new representation for the SCM that uses two distinct sources of information to represent a conjunction of data-dependent balls: a *compression set* to specify the center of each ball and a *message string* to encode the radius value of each ball. Moreover, we have proposed a general data-compression risk bound that

depends explicitly on these two information sources. This bound therefore exhibits a non trivial trade-off between sparsity (the inverse of the compression set size) and the margin (the inverse of the message length) that classifiers should attempt to optimize on the training data. We have also proposed a new learning algorithm for the SCM where the learner can control the amount of trade-off between the sparsity of the classifier and the magnitude of its separating margin. Compared to the algorithm of Marchand and Shawe-Taylor (2002), our experiments on natural data sets indicate that this new learning algorithm generally produces classifiers having a larger separating margin at the expenses of having more balls. The generalization error of classifiers produced by the new algorithm was generally slightly better. Finally, the proposed data-compression risk bound seems to be an effective guide for choosing the proper margin-sparsity trade-off of a classifier.

References

Shai Ben-David and A. Litman. Combinatorial variability of Vapnik-Chervonenkis classes. *Discrete Applied Mathematics*, **86** (1998) 3–25

Kristin P. Bennett. Combining support vector and mathematical programming methods for classifications. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods—Support Vector Learning*, MIT Press, Cambridge MA, (1999) 307–326.

Jinbo Bi, Kristin P. Bennett, Mark Embrechts, Kurt M. Breneman, and Minghu Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Reasearch*, **3** (2003) 1229–1245

Avrim Blum and John Langford. PAC-MDL bounds. In *Proceedings of 16th Annual Conference on Learning Theory, COLT 2003,* Washington, DC, August 2003, volume 2777 of *Lecture Notes in Artificial Intelligence*, Springer, Berlin (2003) 344–357

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, ACM Press, (1992) 144–152

Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21**3** (1995) 269–304

Thore Graepel, Ralf Herbrich, and John Shawe-Taylor. Generalisation error bounds for sparse linear classifiers. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory* (2000) 298–303

Thore Graepel, Ralf Herbrich, and Robert C. Williamson. From margin to sparsity. In *Advances in neural information processing systems 13*, (2001) 210–216

John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Reasearch*, **3** (2005) 273–306.

N. Littlestone and M. Warmuth. Relating data compression and learnability. Technical report, University of California Santa Cruz, Santa Cruz, CA, (1986)

Mario Marchand and John Shawe-Taylor. The set covering machine. *Journal of Machine Learning Reasearch*, **3** (2002) 723–746.

Mario Marchand and Marina Sokolova. Learning with decision lists of data-dependent Features. *Journal of Machine Learning Reasearch*, **6** (2005) 427-451.

S. Mendelson. Rademacher averages and phase transitions in Glivenko-Cantelli class. *IEEE Transactions on Information Theory*, **48** (2002) 251–263

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY (1998)

Ulrike von Luxburg, Olivier Bousquet, and Bernhard Schölkopf. A compression approach to support vector model selection. *Journal of Machine Learning Research*, **5**(2004) 293–323