Mach Learn (2010) 78: 175–201
DOI 10.1007/s10994-009-5137-3

Learning the set covering machine by bound minimization and margin-sparsity trade-off

François Laviolette · Mario Marchand · Mohak Shah · Sara Shanian

Received: 6 November 2007 / Revised: 22 April 2009 / Accepted: 5 July 2009 /
Published online: 25 July 2009
Springer Science+Business Media, LLC 2009

Abstract We investigate classifiers in the sample compression framework that can be specified by two distinct sources of information: a *compression set* and a *message string* of additional information. In the compression setting, a reconstruction function specifies a classifier when given this information. We examine how an efficient redistribution of this reconstruction information can lead to more general classifiers. In particular, we derive risk bounds that can provide an explicit control over *the sparsity of the classifier* and *the magnitude of its separating margin* and a capability to perform a margin-sparsity trade-off in favor of better classifiers. We show how an application to the set covering machine algorithm results in novel learning strategies. We also show that these risk bounds are tighter than their traditional counterparts such as VC-dimension and Rademacher complexity-based bounds that explicitly take into account the hypothesis class complexity. Finally, we show how these bounds are able to guide the model selection for the set covering machine algorithm enabling it to *learn by bound minimization*.

Keywords Set covering machine · Sample compression · Risk bounds · Margin-sparsity trade-off · Bound minimization

Editor: Avrim Blum.

F. Laviolette · M. Marchand · S. Shanian
Department of Computer Science and Software Engineering, Pav. Adrien Pouliot, Laval University,
Quebec, QC, Canada G1V-0A6

F. Laviolette
e-mail: Francois.Laviolette@ift.ulaval.ca

M. Marchand
e-mail: Mario.Marchand@ift.ulaval.ca

S. Shanian
e-mail: Sara.Shanian@ift.ulaval.ca

M. Shah (✉)
Centre for Intelligent Machines, McGill University, Montreal, QC, Canada H3A-2A7
e-mail: Mohak@cim.mcgill.ca

1 Introduction

The sample compression framework (Littlestone and Warmuth 1986; Floyd and Warmuth 1995) has recently been revived and showed significant promise in offering generalization risk bounds for algorithms, that are both tight and usable in practice. The main advantage of these risk bounds is that they do not explicitly depend on some hypothesis class complexity measure (such as VC dimension) when applied to a data-independent hypothesis class.[1] Another important advantage of sample-compression bounds is that they can be applied to data-dependent hypothesis classes. The explicit complexity considerations can be viewed as a limiting factor in other frameworks such as those based on Vapnik-Chervonenkis theory or Rademacher complexities. The hypothesis class complexity dependence in the risk bounds generally make them *tight in asymptotic limits* rendering them unusable for guiding a learning algorithm. Another limitation appears in the form of a lack of control over the quantities that affect the risk bound in explicit terms. Compression bounds on the other hand have a significant advantage in terms of the independence from the explicit inclusion of hypothesis class complexity measure in risk consideration. In this work, we see how we can obtain bounds that enable the user to devise learning strategies that can take advantage of and optimize the quantities affecting the generalization error of the classifier.

Learning algorithms try to produce classifiers with small prediction error by trying to optimize some function that can be computed from a training set of examples and some properties of a classifier. We currently do not know exactly what function should be optimized but several forms have been proposed. At one end of the spectrum, we have the set covering machine (SCM) (Marchand and Shawe-Taylor 2001, 2002), that tries to find the sparsest classifier making few training errors. At the other end, we have the support vector machine (SVM) (Boser et al. 1992), that tries to find the maximum soft-margin separating hyperplane on the training data. Both of these learning machines can produce classifiers having good generalization. The obvious question that arises is: *Is it 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?*

In this work, we investigate this possibility in the compression framework to a generic extent and in particular with regard to the set covering machine learning setting with data-dependent balls. Sample compression algorithms are characterized by the existence of: (i) a compression function that, when given a training set, outputs a small subset of training examples and some additional information that characterize the classifier; and (ii) a reconstruction function that can reconstruct the classifier using this subset and the additional information.

In essence, the (reconstructed) classifier uses two complementary sources of information viz. *the compression set* (the subset of examples output by the compression function) and *the message string* (the additional information needed to obtain a classifier). The main objective of this work is to examine if an efficient distribution of classifier reconstruction information can be done between these two sources to achieve better classification performance. More importantly, we would like to know if it is possible to achieve an explicit

[1] But, if the learning algorithm uses a data-independent hypothesis class, the VC dimension may well affect the size of the compression set. The compression set size and the VC dimension can be closely related. Warmuth (2003) in fact conjectures that for any hypothesis class of VC dimension d, there exist a compression scheme of size at most d (also see Kuzmin and Warmuth 2007; Rubinstein and Rubinstein 2008 for some recent results on specific cases).

control over trading-off the information dispensed between the above two sources in favor of better generalization.

We first consider a generic risk bound in the compression framework and derive a form that shows explicit dependence in terms of a prior distribution on the compression set and the associated messages. We then go on to investigate if the information can be more efficiently dispensed between these two quantities.

Our first approach is an information theoretic approach inspired by the PAC-MDL approach of Blum and Langford (2003) who have derived a PAC-MDL risk bound for classifiers that unifies most of the standard risk bounds in one common framework. This PAC-MDL bound is stated in a non-standard "transductive" setting where, given a training set of m labeled examples, the goal of the learner is to construct a small message string that can be used by a receiver to predict the labels of an unlabeled set of $m+n$ examples that contains the m training examples (without their labels). Consequently, one important drawback of the PAC-MDL bound is that it does not capture the sample-compression bound (Littlestone and Warmuth 1986) very well. Indeed, in the PAC-MDL transductive setting, the learner has to build a message string that specifies a compression subset among $m+n$ examples whereas, in the usual "inductive" setting,[2] the learner just needs to specify the compression subset among m training examples. Because of this, the PAC-MDL bound is usually substantially larger than the sample-compression bound.

In Sect. 2 of this paper, we therefore propose a generic data-compression risk bound that, although less universal than the PAC-MDL bound, unifies the Occam's razor bound (Blumer et al. 1987) and the sample-compression bound in the usual inductive setting. This bound, as we will see, is a tighter version of the sample-compression bound of Littlestone and Warmuth (1986). The bound reduces to the tightest version of the Occam's razor bound (Langford 2005) when no compression set is used and also reduces to the tightest version of the sample-compression bound when no message string of additional information is used. We illustrate, on the set covering machine (SCM) (Marchand and Shawe-Taylor 2002), how the learner can tradeoff these two complementary sources of information (the compression set and the message string) to obtain classifiers having a smaller risk.

Our second approach is a PAC-Bayes approach. The PAC-Bayes theorem was first proposed by McAllester (2003) and then improved by others (see Langford 2005 for a survey). However, for all these versions of the PAC-Bayes theorem, the prior P must be defined without reference to the training data. Consequently, these theorems cannot be applied to the sample-compression setting where classifiers are partly described by a subset of the training data (as for the case of the SCM). We draw motivation from the work of Laviolette and Marchand (2007) who have now generalized the PAC-Bayes approach to the sample-compression setting. It should be noted that, in this work, we adopt the PAC-Bayes approach only for the message portion (for a given fixed compression set) and derive a bound that is valid uniformly for all compression subsets. Consequently, the bound derivation presented here is simpler and more specialized than the one presented in Laviolette and Marchand (2007). However, the latter bound is valid for the more general case of a stochastic average over several compression subsets.

We then go on to show how these two approaches yield new learning strategies for the SCM algorithm. This is an extension of the works in Laviolette et al. (2005, 2006), Shah (2006). Basically, in order to incorporate the notion of margin, we use two approaches. The first one is an information theoretic approach that uses a bit string to code for the separating

[2] In this setting, the task of the learner is to find a classifier with the smallest true risk.

margin. The second approach aims at obtaining an actual margin interval around the decision surface of the classifier.

1.1 Organization

In Sect. 2, we derive the generic compression risk bound that shows explicit dependence on the compression set and a message string of additional information. For completeness, Sect. 3 gives a brief overview of the set covering machine algorithm. Then in Sect. 4, we first show how we can recover the compression bound of the original SCM formulation and then go on to show how we can redistribute the reconstruction information efficiently resulting in a bound that can perform a non-trivial margin-sparsity trade-off. Section 5 then gives a learning strategy inspired by the new bound derived in the previous section. The PAC-Bayes bound is then derived in Sect. 6 for which a soft-greedy learning algorithm is proposed in Sect. 7.

An empirical analysis of the performance of various approaches and their comparison with the SVM appear in Sect. 8 where we also investigate if the risk bounds by themselves can guide the learning process. Section 9 then places the findings in context and provides an unified view of this work. Finally, we conclude in Sect. 10.

2 A generic data compression risk bound

In this section, we give a tight data-compression risk bound on the generalization error of a sample compressed classifier. The bound takes into account, along with the empirical performance of the classifier, the compression achieved by the classifier and the additional information required for reconstruction in the form of associated messages. In essence, the classifier, as mentioned above, is signified by two sources of information: *the compression set* and the *message string*. Dispensing the information content between these two quantities efficiently will enable us to obtain better trade-off capabilities between the sparsity of a classifier and the magnitude of its separating margin as we will see later.

Let $\mathbf{z}$ be a random tuple representing an example-label pair, i.e. $\mathbf{z} = (\mathbf{x}, y)$ such that $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$. We consider classification problems where the input space $\mathcal{X}$ consist of an arbitrary subset of $\mathbb{R}^n$ and the output space $\mathcal{Y} = \{0, 1\}$. Further, 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 (true) risk $R(f)$ of any classifier f is defined as the probability that it misclassifies an example drawn according to D:

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

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

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

Let $\mathbf{Z}^m$ denote the collection of m random variables $\{\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_m\}$ whose instantiations gives the training set $S = \mathbf{z}^m = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$.

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 samples, 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 sample $S = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$, we define the compression set $\mathbf{z_i}$ by a vector of indices $\mathbf{i}$:

$$\mathbf{i} \stackrel{\text{def}}{=} (i_1, i_2, \ldots, i_{|\mathbf{i}|}) \tag{1}$$

$$\text{with} \quad i_j \in \{1, \ldots, m\} \quad \forall j \tag{2}$$

$$\text{and} \quad i_1 < i_2 < \cdots < i_{|\mathbf{i}|} \tag{3}$$

where $|\mathbf{i}|$ denotes the number of indices present in $\mathbf{i}$. In addition to the notation used so far, we will use $\bar{\mathbf{i}}$ to denote the set of indices not present in $\mathbf{i}$. Hence, we have $S = \mathbf{z_i} \cup \mathbf{z_{\bar{i}}}$ for any vector $\mathbf{i} \in \mathcal{I}$ where $\mathcal{I}$ denotes the set of the 2^m possible realizations of $\mathbf{i}$.

When given an arbitrary compression set $\mathbf{z_i}$ and an arbitrary information message σ, the *reconstruction function* $\mathcal{R}$ must output a classifier that we will denote by $\mathcal{R}(\sigma, \mathbf{z_i})$. The information message σ is chosen from the set $\mathcal{M}(\mathbf{z_i})$. This set $\mathcal{M}(\mathbf{z_i})$ consists of all the distinct messages that can be attached to the compression set $\mathbf{z_i}$. Further, $\mathcal{M}(\mathbf{z_i})$ must be defined *a priori* (before observing S) for all possible compression sets $\mathbf{z_i}$. We denote by $\mathcal{M}_S$ the union of all such sets of messages:

$$\mathcal{M}_S \stackrel{\text{def}}{=} \bigcup_{\mathbf{i} \in \mathcal{I}} \mathcal{M}(\mathbf{z_i})$$

This should be contrasted with both the pure sample compression setting where the classifier can be reconstructed solely from the compression set and the usual data-independent settings where the classifier space is defined without reference to the training data. 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, 2005). Indeed, in the case of the perceptron learning rule, the compression set consists of the training examples on which the perceptron rule updates its weights while learning. In the case of the SVM, the compression set consists of all the support vectors identified by the learning algorithm. In both the cases the learning algorithms themselves acts as the reconstruction functions when applied to the respective compression sets. No additional information is required when the classifiers are reconstructed in this manner.

On the other hand, the usual data-independent setting specifies, for learning algorithms, the space of classifiers $\mathcal{H}$ without reference to the training data. We can recover this usual setting when each classifier is identified only by a message σ taken from $\mathcal{M}(\mathbf{z_i})$ for $\mathbf{z_i} = \emptyset$. In this case the reconstructed classifier is of the form $\mathcal{R}(\sigma, \emptyset)$. Hence, in this limit, we have a data-independent set $\mathcal{H}$ of classifiers given by the reconstruction function $\mathcal{R}$ and the set of messages $\mathcal{M}$ such that

$$\mathcal{H} = \{\mathcal{R}(\sigma, \emptyset) | \sigma \in \mathcal{M}\}$$

The reconstruction function for SCMs needs both a compression set and a message string. We define priors over $\mathcal{I} \times \mathcal{M}_S$ for any possible $S \in D^m$. Moreover, for any given S, we will consider only the priors P_S that can be factored as

$$P_S(\mathbf{i}, \sigma) = P_{\mathcal{I}}(\mathbf{i}) P_{\mathcal{M}(\mathbf{z_i})}(\sigma)$$

where $P_{\mathcal{I}}(\mathbf{i})$ is the prior probability of using the vector $\mathbf{i}$ of indices as defined above and where $P_{\mathcal{M}(\mathbf{z_i})}(\sigma)$ is the prior probability of using the message string σ given that we use the compression set $\mathbf{z_i}$. The message string σ can also be a *parameter*[3] chosen from a continuous set $\mathcal{M}(\mathbf{z_i})$. In this case, $P_{\mathcal{M}(\mathbf{z_i})}(\sigma)$ would specify a probability density function.

Later, we will see how the learner can tradeoff the compression set size with the length of the message strings 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. To obtain the tightest possible risk bound, we fully exploit the fact that the distribution of classification errors is a binomial. The binomial tail $\text{Bin}(k, m, r)$ 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:

$$\text{Bin}(k, m, r) \overset{\text{def}}{=} \sum_{i=0}^{k} \binom{m}{i} r^i (1-r)^{m-i}$$

Following (Blum and Langford 2003; Langford 2005), we now define the *binomial tail inversion* $\overline{\text{Bin}}(k, m, \delta)$ 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:

$$\overline{\text{Bin}}(k, m, \delta) \overset{\text{def}}{=} \sup\{r : \text{Bin}(k, m, r) \geq \delta\}$$

From this definition, it follows that $\overline{\text{Bin}}(mR_S(f), m, \delta)$ 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:

$$\mathbf{P}_{\mathbf{Z}^m}\{R(f) \leq \overline{\text{Bin}}(mR_{\mathbf{Z}^m}(f), m, \delta)\} \geq 1-\delta \quad \forall f \tag{4}$$

where we denote $\Pr_{\mathbf{z}^m \sim D^m}(.)$ by $\mathbf{P}_{\mathbf{Z}^m}(.)$.

The bound $\overline{\text{Bin}}(mR_S(f), m, \delta)$ does not hold *simultaneously* (i.e., uniformly) for all classifiers f member of some predefined class $\mathcal{F}$. As a result the quantifier $\forall f$ appears *outside* the probability $\mathbf{P}_{\mathbf{Z}^m}\{\cdot\}$. In contrast, the proposed risk bound of Theorem 1 holds uniformly for all compression sets and message strings. The above bound is sometimes called a *test set upper bound*, since it can be used with a test set to give an upper bound for any fixed classifier (Langford 2005).

The proposed risk bound (Theorem 1) 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 the message string. It also has the property to reduce to the Occam's Razor bound when the sample compression set vanishes. The idea of using a message string as an additional source of information was also used in (Littlestone and Warmuth 1986; Ben-David and Litman 1998) to obtain a sample-compression bound looser than the bound presented here.

Moreover, the proposed bound applies to any compression set-dependent distribution of messages $P_{\mathcal{M}(\mathbf{z_i})}$ satisfying:

$$\sum_{\sigma \in \mathcal{M}(\mathbf{z_i})} P_{\mathcal{M}(\mathbf{z_i})}(\sigma) \leq 1 \quad \forall \mathbf{z_i} \tag{5}$$

[3] In the present case of SCM, this would correspond to the radii values for the data-dependent balls as we will see later.

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

$$\sum_{\mathbf{i}\in\mathcal{I}} P_{\mathcal{I}}(\mathbf{i}) \leq 1 \tag{6}$$

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 $P_{\mathcal{I}}$ on the set 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:*

$$\begin{aligned}&\mathbf{P}_{\mathbf{Z}^m}\Big\{\forall \mathbf{i} \in \mathcal{I}, \forall \sigma \in \mathcal{M}(\mathbf{Z_i}):\ R(\mathcal{R}(\sigma, \mathbf{Z_i}))\\ &\quad \leq \overline{\mathrm{Bin}}\big((m - |\mathbf{i}|)R_{\mathbf{Z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z_i})), (m - |\mathbf{i}|), P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta\big)\Big\} \geq 1 - \delta\end{aligned}$$

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 $\mathbf{z_i}$.

Proof Consider:

$$\begin{aligned}P' &\overset{\text{def}}{=} \mathbf{P}_{\mathbf{Z}^m}\Big\{\exists \mathbf{i} \in \mathcal{I}:\ \exists \sigma \in \mathcal{M}(\mathbf{Z_i}):\ R(\mathcal{R}(\sigma, \mathbf{Z_i}))\\ &\quad > \overline{\mathrm{Bin}}\big((m - |\mathbf{i}|)R_{\mathbf{Z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z_i})), m - |\mathbf{i}|), P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta\big)\Big\}\end{aligned}$$

To prove the theorem, we show that $P' \leq \delta$. Since $\mathbf{P}_{\mathbf{Z}^m}(\cdot) = \mathbf{E}_{\mathbf{Z_i}}\mathbf{P}_{\mathbf{Z}_{\bar{\mathbf{i}}}|\mathbf{Z_i}}(\cdot)$, and since the examples are supposed i.i.d., (4) applies when we replace $\mathbf{Z}^m$ and m by $\mathbf{z}_{|\bar{\mathbf{i}}|}|\mathbf{z_i}$ and $m - |\mathbf{i}|$ respectively. This, together with the union bound, and (5) and (6) imply that we have:

$$\begin{aligned}P' &\leq \sum_{\mathbf{i}\in\mathcal{I}} \mathbf{E}_{\mathbf{Z_i}} \sum_{\sigma\in\mathcal{M}(\mathbf{Z_i})} \mathbf{P}_{\mathbf{Z}_{\bar{\mathbf{i}}}|\mathbf{Z_i}}\Big\{R(\mathcal{R}(\sigma, \mathbf{Z_i}))\\ &\quad > \overline{\mathrm{Bin}}\big((m - |\mathbf{i}|)R_{\mathbf{Z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z_i})), (m - |\mathbf{i}|), P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta\big)\Big\}\\ &\leq \sum_{\mathbf{i}\in\mathcal{I}} \mathbf{E}_{\mathbf{Z_i}} \sum_{\sigma\in\mathcal{M}(\mathbf{Z_i})} P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta\\ &\leq \delta\end{aligned}$$

□

The proof of Theorem 1 contains three inequalities. The last two inequalities come from (4), (5), and (6) 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 for all possible choices of message strings given a compression set.

It is important to note that, once $P_{\mathcal{I}}$ and $P_{\mathcal{M}(\mathbf{z_i})}$ are specified, the risk bound of Theorem 1 for classifier $\mathcal{R}(\sigma, \mathbf{z_i})$ depends on its empirical risk *and* on the product $P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)$. However, $\ln(\frac{1}{P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)})$ is just the amount of information needed to specify a classifier $\mathcal{R}(\sigma, \mathbf{z_i})$ once we are given a training set and the priors $P_{\mathcal{I}}$ and $P_{\mathcal{M}(\mathbf{z_i})}$. 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 $\ln(1/P_{\mathcal{M}(\mathbf{z_i})}(\sigma))$ term is the information content of the message string σ given $\mathbf{z_i}$. Consequently the bound of Theorem 1 specifies quantitatively how much training error 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. Here, we make use of the following approximations (proof provided in Appendix A) for the binomial tail inversion:

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

$$\begin{aligned}\overline{\mathrm{Bin}}(k,m,\delta) &\leq 1-\exp\left(\frac{-1}{m-k}\left[\ln\binom{m}{k}+\ln\left(\frac{1}{\delta}\right)\right]\right)\\ &\leq \frac{1}{m-k}\left[\ln\binom{m}{k}+\ln\left(\frac{1}{\delta}\right)\right] \qquad (7)\end{aligned}$$

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*:

$$\begin{aligned}&\mathbf{P}_{\mathbf{Z}^m}\Bigg\{\forall \mathbf{i} \in \mathcal{I}, \forall \sigma \in \mathcal{M}(\mathbf{Z_i}):\ R(\mathcal{R}(\sigma,\mathbf{Z_i}))\\ &\quad\leq 1-\exp\left(\frac{-1}{m-d-k}\left[\ln\binom{m-d}{k}+\ln\left(\frac{1}{P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta}\right)\right]\right)\Bigg\} \geq 1-\delta \qquad (8)\end{aligned}$$

and, *consequently*:

$$\begin{aligned}&\mathbf{P}_{\mathbf{Z}^m}\Bigg\{\forall \mathbf{i} \in \mathcal{I}, \forall \sigma \in \mathcal{M}(\mathbf{Z_i}):\ R(\mathcal{R}(\sigma,\mathbf{Z_i}))\\ &\quad\leq \frac{1}{m-d-k}\left[\ln\binom{m-d}{k}+\ln\left(\frac{1}{P_{\mathcal{I}}(\mathbf{i})P_{\mathcal{M}(\mathbf{z_i})}(\sigma)\delta}\right)\right]\Bigg\} \geq 1-\delta \qquad (9)\end{aligned}$$

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 and when σ is short. 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 naturally to our case.

The bound of (8) is very similar to, and slightly tighter than, the recent bound of Marchand and Sokolova (2005) owing to the more efficient treatment of errors by the binomial tail inversion.

The looser bound of (9) 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:

$$P_{\mathcal{M}(\mathbf{z_i})}(\sigma) = 1/|\mathcal{M}| \quad \forall \sigma \in \mathcal{M} \tag{10}$$

$$P_{\mathcal{I}}(\mathbf{i}) = \binom{m}{|\mathbf{i}|}^{-1} (m+1)^{-1} \quad \forall \mathbf{i} \in \mathcal{I} \tag{11}$$

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

$$P_{\mathcal{I}}(\mathbf{i}) = \binom{m}{|\mathbf{i}|}^{-1} \zeta(|\mathbf{i}|) \quad \text{with } \zeta(a) \overset{\text{def}}{=} \frac{6}{\pi^2}(a+1)^{-2} \; \forall a \in \mathbb{N} \tag{12}$$

which satisfies the constraint of (6) 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.

3 The set covering machine

The SCM algorithm was motivated originally by the idea of learning a conjunction (or disjunction) of literals via the standard monomial learning algorithm proposed by Valiant (1984). Haussler (1988) showed that this problem could be reduced to the minimum set cover problem which, although NP-hard, has a good worst-case upper bound for the greedy heuristic (Chvátal 1979). Motivated by this observation Marchand and Shawe-Taylor (2001, 2002) generalized this algorithm for learning conjunctions (or disjunctions) of data-dependent Boolean attributes to the case of learning these functions over arbitrary sets of Boolean valued features, i.e. features constructed from data. Also, the algorithm provides some learning parameters to control the tradeoff between the accuracy and the size of the conjunction (or disjunction) so as to deal with the problems of noisy data and overfitting.

Let the training set $S = \mathcal{P} \cup \mathcal{N}$ consists of a set $\mathcal{P}$ of positive training examples and a set $\mathcal{N}$ of negative training examples. A *feature* is defined as an arbitrary Boolean-valued function that maps $\mathcal{X}$ onto $\{0, 1\}$.

Let $\digamma = \{h_i\}_{i=1}^{|\digamma|}$ be any set of features h_i. The learning algorithm when given any such set $\digamma$ returns a small subset $\mathcal{F} \subset \digamma$ of features. Given this subset $\mathcal{F}$ and an arbitrary input vector $\mathbf{x} \in \mathcal{X}$, the output $f(\mathbf{x})$ of the SCM is defined to be:

$$f(\mathbf{x}) = \begin{cases} \bigvee_{i \in \mathcal{F}} h_i(\mathbf{x}) & \text{for a disjunction} \\ \bigwedge_{i \in \mathcal{F}} h_i(\mathbf{x}) & \text{for a conjunction} \end{cases}$$

where $h_i(\mathbf{x}) \in \{0, 1\}$ denotes the output of feature h_i on $\mathbf{x}$.

We will use the usual definition for *consistency*:

Definition 1 A function (or a feature) is said to be consistent with an example if it correctly classifies that example. Similarly, a function (or a feature) is said to be consistent with a set of examples if it correctly classifies all the examples in that set.

Here, we will focus on conjunction case since the case of disjunction is completely analogous. From the above definition, it follows that f is consistent with $\mathcal{P}$ iff each $h_i \in \mathcal{F}$ is consistent with $\mathcal{P}$. Moreover, if Q_i denotes the subset of examples of $\mathcal{N}$ on which feature h_i makes no errors, then f makes no error on $\mathcal{N}$ if and only if $\bigcup_{i \in \mathcal{F}} Q_i = \mathcal{N}$. Hence, as was first observed by Haussler (1988), the problem of finding the smallest set $\mathcal{F}$ for which f makes no training errors is just the problem of finding the smallest collection of Q_is that cover all $\mathcal{N}$ (where each corresponding h_i makes no error on $\mathcal{P}$). This is the well-known *minimum set cover problem* (Garey and Johnson 1979). The interesting fact is that, although it is NP-hard to find the smallest cover, the *set covering greedy algorithm* will always find a cover of size at most $z \ln(|\mathcal{N}|)$ when the smallest cover that exists is of size z (Chvátal 1979; Kearns and Vazirani 1994). Moreover this algorithm is very simple to implement and just consists of the following steps: first choose the set Q_i which covers the largest number of elements in $\mathcal{N}$, remove from $\mathcal{N}$ and each Q_j the elements that are in Q_i, then repeat this process of finding the set Q_k of largest cardinality and updating $\mathcal{N}$ and each Q_j until there are no more elements in $\mathcal{N}$.

The SCM built on the features found by the set covering greedy algorithm will be consistent with S only when there exists a subset $\mathcal{E} \subset \mathcal{F}$ of features whose conjunction (or a disjunction) is consistent with S. However, this constraint is not really required in practice since we do want to permit the user of a learning algorithm to control the trade-off between the accuracy achieved on the training data and the complexity (here the size) of the classifier. Indeed, a small SCM which makes a few errors on the training set might give better generalization than a larger SCM (with more features) which makes zero training errors. One way to include this flexibility into the SCM is to stop the set covering greedy algorithm when a maximum number v of features is reached. In this case, the SCM will contain fewer features and will make errors on those training examples that are not covered. But these examples all belong to $\mathcal{N}$ and, in general, we do need to be able to make errors on training examples of both classes. Hence, early stopping is generally not sufficient and, in addition, we need to consider features that also make some errors with $\mathcal{P}$ provided that many more examples in $\mathcal{N}$ can be covered. Hence, for a feature h, let us denote by Q_h the set of examples in $\mathcal{N}$ covered by feature h and by R_h the set of examples in $\mathcal{P}$ on which h makes an error. Given that each example in $\mathcal{P}$ misclassified by h should decrease by some fixed *penalty* p its "importance", the *usefulness* U_h of feature h is defined by the following equation:

$$U_h \stackrel{\text{def}}{=} |Q_h| - p \cdot |R_h| \tag{13}$$

Hence, the set covering greedy algorithm is modified in the following way. Instead of using the feature that covers the largest number of examples in $\mathcal{N}$, the feature $h \in \mathcal{F}$ that has the highest usefulness value U_h is used. We remove from $\mathcal{N}$ and each Q_g (for $g \neq h$) the elements that are in Q_h and we remove from each R_g (for $g \neq h$) the elements that are in R_h. Note that we update each such set R_g because a feature g that makes an error on an example in $\mathcal{P}$ does not increase the error of the machine if another feature h is already making an error on that example. We repeat this process of finding the feature h of largest usefulness U_h and updating $\mathcal{N}$, and each Q_g and R_g, until only an ϵ fraction of elements remain in $\mathcal{N}$ (early stopping the greedy).

3.1 Data-dependent balls

Marchand and Shawe-Taylor (2001, 2002) gave an implementation of the SCM algorithm with a set of features they called *data-dependent balls* and proposed a risk bound for SCM with this set of features. We will use this set of features for our case too. In this 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:

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

where $\neg y_c$ is the boolean complement of y_c.

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 $\rho = d(\mathbf{x}_c, \mathbf{x}_b)$. Moreover, only the positive examples are used to denote a border point although both a negative as well as a positive example can be a potential ball center. This is done to ensure the consistency of each feature in the conjunction to the set of positive examples in the compression set so as to optimize the use if messages (since we do not need additional information for border points that are negative examples).[4] A data-dependent ball centered at $\mathbf{x}_c$ has its radius defined as $d(\mathbf{x}_c, \mathbf{x}_b) - \epsilon$ when $\mathbf{x}_c$ is a negative example and as $d(\mathbf{x}_c, \mathbf{x}_b) + \epsilon$ when $\mathbf{x}_c$ is a positive example, and where ϵ is a arbitrarily small positive real number.

In the next section, we show how we can apply the risk bounds of Theorem 1 and Corollary 1 to derive the risk bounds for the SCM with data-dependent balls. First, we will show how we can recover the original sample compression bound (though tighter in our case) for the SCM by incorporating most of the classifier reconstruction information in the compression set. Then, we will go on to describe how an efficient redistribution of the reconstruction information between the compression set and the corresponding message distribution can lead to classifiers that offer explicit control over trading-off magnitude of the separating margin and the sparsity.

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 (10). 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.

4 Adapting the generic risk bound

4.1 Recovering the compression bound for SCM

Let us now see how, incorporating most of the classifier information in the compression set leads to a (relatively) pure compression bound. We will basically recover (a tighter version of) the compression bound for the set covering machine.

Consider the formulation of the set covering machine with data-dependent balls described in Sect. 3.1. In order to understand the distribution of messages that we need to specify for this case, let us start with the reconstruction function in the case of the classical SCM. For this case, we can define the reconstruction function $\mathcal{R}(\mathbf{z_i}, \sigma)$ as follows (Marchand and Shawe-Taylor 2002). Given any arbitrary compression set $\mathbf{z_i} = \mathbf{z_i}^{C_p} \cup \mathbf{z_i}^{C_n} \cup \mathbf{z_i}^{B_p}$

[4] In an analogous manner, in the case of disjunction, we require that each feature be consistent with the negative examples in the compression set.

where $\mathbf{z}_{\mathbf{i}}^{C_p}$ and $\mathbf{z}_{\mathbf{i}}^{C_n}$ denote the subsets containing the positive and negative center examples respectively and $\mathbf{z}_{\mathbf{i}}^{B_p}$ denotes the set of positive border examples. Then, for each $\mathbf{x}_i \in \mathbf{z}_{\mathbf{i}}^{C_p}$, $\mathcal{R}$ creates a ball centered at $\mathbf{x}_i$ with radius $\rho = \max_{j \in \mathbf{z}_{\mathbf{i}}^{C_p} \cup \mathbf{z}_{\mathbf{i}}^{B_p}} d(\mathbf{x}_i, \mathbf{x}_j) + \epsilon$. Similarly, $\mathcal{R}$ creates a ball centered at each $\mathbf{x}_i \in \mathbf{z}_{\mathbf{i}}^{C_n}$ with radius $\rho = \min_{j \in \mathbf{z}_{\mathbf{i}}^{C_p} \cup \mathbf{z}_{\mathbf{i}}^{B_p}} d(\mathbf{x}_i, \mathbf{x}_j) - \epsilon$. Finally, $\mathcal{R}$ builds a conjunction of these features. Note therefore that the message σ given to $\mathcal{R}$ only has to give the necessary information to reconstruct the set $\mathbf{z}_{\mathbf{i}}^{B_p}$. In other words, σ simply points out which elements of $\mathbf{z}_{\mathbf{i}}^{C_p}$ are used at least once as border points.

Let $n(\mathbf{z}_{\mathbf{i}}) = |\mathbf{z}_{\mathbf{i}}^{C_n}|$ and $p(\mathbf{z}_{\mathbf{i}}) = |\mathbf{z}_{\mathbf{i}}^{C_p}| + |\mathbf{z}_{\mathbf{i}}^{B_p}|$ be, respectively, the number of negative and the number of positive examples in compression set $\mathbf{z}_{\mathbf{i}}$. Let $b(\sigma)$ be the number of border point examples specified in message σ and let $\zeta(a)$ be the same as defined in (12). We then need a message to identify the border points from among the set of positive examples in compression set. Since, we do not have any *a priori* knowledge about which positive examples in the compression set are border points, we assign equal probability to all possible subsets of size $b(\sigma)$ in $p(\mathbf{z}_{\mathbf{i}})$. Hence, we establish the following message distribution in terms of the border examples to code radii:

$$P_{\mathcal{M}(\mathbf{Z}_{\mathbf{i}})}(\sigma) = \zeta(b(\sigma)) \cdot \binom{p(\mathbf{z}_{\mathbf{i}})}{b(\sigma)}^{-1} \tag{14}$$

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

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

With this distribution $P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}$, we obtain the following sample compression risk bound.

Corollary 2 *Let $\mathcal{R}$ be the reconstruction function for the SCM as described above. For any prior distribution $P_{\mathcal{I}}$ of vectors of indices, for any compression set-dependent distribution of messages $P_{\mathcal{M}(\mathbf{z}_{\mathbf{i}})}$, and for any $\delta \in (0, 1]$, we have:*

$$\mathbf{P}_{\mathbf{Z}^m}\Bigg\{\forall \mathbf{i} \in \mathcal{I}, \forall \sigma \in \mathcal{M}(\mathbf{Z}_{\mathbf{i}}) \colon \; R(\mathcal{R}(\sigma, \mathbf{Z}_{\mathbf{i}})) \leq \overline{\mathrm{Bin}}\left((m - |\mathbf{i}|) R_{\mathbf{Z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z}_{\mathbf{i}})), (m - |\mathbf{i}|), \frac{\zeta(|\mathbf{i}|) \;\; \zeta(b(\sigma)) \cdot \delta}{\binom{m}{|\mathbf{i}|} \cdot \binom{p(\mathbf{z}_{\mathbf{i}})}{b(\sigma)}}\right)\Bigg\} \geq 1 - \delta$$

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 $\mathbf{z}_{\mathbf{i}}$, $p(\mathbf{z}_{\mathbf{i}})$ denotes the number of positive examples in $\mathbf{z}_{\mathbf{i}}$ and $b(\sigma)$ denote the number of positive examples that are the border points.

Note that the risk bound of Corollary 2 is tighter than the one provided by Marchand and Shawe-Taylor (2002) owing to the more efficient treatment of the training errors by the virtue of the binomial tail inversion.

4.2 Margins in terms of message distribution

Let us now see how we can distribute the classifier reconstruction information more efficiently between the *compression set* and *the message string* so as to get not only tighter guarantees but also a capability to perform a non-trivial margin-sparsity trade-off.

Consider an alternate scheme to encode the radii values based on utilizing a bit string. The idea is to use a message string having the fewest number of bits, much like an Occam's Razor framework. 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 can find a message strings that can code each radius value efficiently, i.e., using only a few bits. 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, the problem is reduced to one 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.[5] 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 (for some receiver). It only depends on the a priori probabilities assigned to each possible realization of σ. We choose the following distribution:

$$\begin{aligned} P_{\mathcal{M}(\mathbf{Z_i})}(\sigma) &\stackrel{\text{def}}{=} P_{\mathcal{M}(\mathbf{Z_i})}(l_1, s_1, \ldots, l_{|\mathbf{i}|}, s_{|\mathbf{i}|}) \\ &= \prod_{i \in \mathbf{i}} \zeta(l_i) \cdot 2^{-l_i} \end{aligned} \tag{15}$$

where $\zeta(l_i)$ is given by (12).

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 further too rapidly as l_i increases. Other choices are clearly possible.

With the above choice for the message distribution, we deduce the following bound:

[5] 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).

Corollary 3 *Let $\mathcal{R}$ be the reconstruction function for the new SCM as described above. 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*:

$$\mathbf{P}_{\mathbf{Z}^m}\Bigg\{\forall \mathbf{i} \in \mathcal{I}, \forall \sigma \in \mathcal{M}(\mathbf{Z_i}) : \, R(\mathcal{R}(\sigma, \mathbf{Z_i})) \leq \overline{\mathrm{Bin}}\left((m - |\mathbf{i}|) R_{\mathbf{Z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\sigma, \mathbf{Z_i})), (m - |\mathbf{i}|), \frac{\zeta(|\mathbf{i}|)}{\binom{m}{|\mathbf{i}|}} \left(\prod_{i \in \mathbf{i}} \zeta(l_i) \cdot 2^{-l_i} \right) \delta \right) \Bigg\} \geq 1 - \delta$$

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 $\mathbf{z_i}$.

By comparing the risk bounds of Corollaries 2 and 3 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 $m \lessapprox R/(r_2 - r_1)$.

Let us emphasize that the risk bound of Corollary 3, provides a guide for choosing the appropriate tradeoff between sparsity (the inverse of the size of the compression set) and margin (represented here by the inverse of the expected 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).

5 Learning in the premise of Corollary 3

In this section, we propose an alternate learning strategy for the SCM that can exploit the capabilities of the bound of Corollary 3. We will also examine how good this bound is in guiding the model selection process.

Ideally, we would like to find a conjunction of balls that minimizes the risk bound of Corollary 3. Unfortunately, this cannot be done efficiently in all cases since this problem is at least as hard as the (NP-hard) minimum set cover problem (Marchand and Shawe-Taylor 2002) as discussed before. Hence, we can make use of the *set covering greedy heuristic* of Sect. 3. However, we need to adapt this heuristic to the new framework. We do this in the following way.

We first modify the greedy heuristic 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 (15) 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 Sect. 4.2 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 (15) by $1/t$. Nevertheless, this introduces one more adjustable parameter in the learning algorithm: the value of R.[6] Therefore, p, l^*, and R are the "learning parameters", in addition to the early stopping criterion v (denoting the maximum number of features allowed) that our heuristic uses to generate a set of classifiers. At the end, we can use the bound of Corollary 3 to select the best classifier. Another alternative is to determine the best parameter values by cross-validation.

6 A PAC-Bayes risk bound

In the learning strategy for SCM motivated by Corollary 3, we achieved efficient control over trading-off the magnitude of the separating margin and the sparsity of the classifier. However, it did impose a scaling constraint. Consequently, the proposed algorithm for the SCM suffered from the fact that the radius values, used in the final classifier, depends on an *a priori* chosen distance scale R. In this section, we apply a PAC-Bayes approach to code the messages given a fixed compression set in an attempt to alleviate the scaling constraint imposed earlier. We derive a PAC-Bayes risk bound that is valid uniformly for all compression sets.

Recall that, given a training set S, the compression set $\mathbf{z_i} \subseteq S$ is defined by a vector of indices $\mathbf{i} \overset{\text{def}}{=} (i_1, \dots, i_{|\mathbf{i}|})$ that points to individual examples in S. For the case of a conjunction of balls, each $j \in \mathbf{i}$ points to a training example that is used for a ball center and the message string σ in this setting will be the vector $\boldsymbol{\rho}$ of radius values that are used for the balls. Hence, given $\mathbf{z_i}$ and $\boldsymbol{\rho}$, the classifier (i.e. conjunction) is obtained from $\mathcal{R}(\boldsymbol{\rho}, \mathbf{z_i})$.[7]

Recently, Laviolette and Marchand (2007) have extended the PAC-Bayes theorem to the sample-compression setting. Their proposed PAC-Bayes risk bound depends on a data-independent prior P and a data-dependent posterior Q that are both defined on $\mathcal{I} \times \mathcal{M}$ where $\mathcal{I}$ denotes the set of the 2^m possible index vectors $\mathbf{i}$ and $\mathcal{M}$ denotes, in our case, the set of possible radius vectors $\boldsymbol{\rho}$. The posterior Q is used by a stochastic classifier, called the *sample-compressed Gibbs classifier* G_Q, defined as follows. Given a training set S and given a new (testing) input example $\mathbf{x}$, a sample-compressed Gibbs classifier G_Q chooses randomly $(\mathbf{i}, \boldsymbol{\rho})$ according to Q to obtain classifier $\mathcal{R}(\boldsymbol{\rho}, \mathbf{z_i})$ which is then used to determine the class label of $\mathbf{x}$.

Here, we focus on the case where, given any training set S, the learner returns a Gibbs classifier defined with a posterior distribution Q having all its weight on a single vector $\mathbf{i}$. Hence, a single compression set $\mathbf{z_i}$ will be used for the final classifier. However, the radius ρ_i for each $i \in \mathbf{i}$ will be chosen stochastically according to the posterior Q. Hence we consider posteriors Q such that $Q(\mathbf{i}', \boldsymbol{\rho}) = I(\mathbf{i} = \mathbf{i}')Q_{\mathbf{i}}(\boldsymbol{\rho})$ where $\mathbf{i}$ is the vector of indices chosen by the learner. Hence, given a training set S, the true risk $R(G_{Q_\mathbf{i}})$ of $G_{Q_\mathbf{i}}$ and its empirical risk $R_S(G_{Q_\mathbf{i}})$ are defined by

$$R(G_{Q_\mathbf{i}}) \overset{\text{def}}{=} \mathop{\mathbf{E}}_{\boldsymbol{\rho} \sim Q_\mathbf{i}} R(\mathcal{R}(\boldsymbol{\rho}, \mathbf{z_i})); \qquad R_S(G_{Q_\mathbf{i}}) \overset{\text{def}}{=} \mathop{\mathbf{E}}_{\boldsymbol{\rho} \sim Q_\mathbf{i}} R_{\mathbf{z}_{\bar{\mathbf{i}}}}(\mathcal{R}(\boldsymbol{\rho}, \mathbf{z_i}))$$

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

[7] We assume that the examples in $\mathbf{z_i}$ are ordered as in S so that the kth radius value in $\boldsymbol{\rho}$ is assigned to the kth example in $S_\mathbf{i}$.

where $\bar{\mathbf{i}}$ denotes the set of indices not present in $\mathbf{i}$. Thus, $\bar{\mathbf{i}} \cap \mathbf{i} = \emptyset$ and $\mathbf{i} \cup \bar{\mathbf{i}} = (1, \ldots, m)$.

In contrast with the posterior Q, the prior P assigns a non zero weight to several vectors $\mathbf{i}$. Let $P_{\mathcal{I}}(\mathbf{i})$ denote the prior probability P assigned to vector $\mathbf{i}$ and let $P_{\mathbf{i}}(\boldsymbol{\rho})$ denote the probability density function associated with prior P given $\mathbf{i}$. The risk bound depends on the Kullback-Leibler divergence $\mathrm{KL}(Q \| P)$ between the posterior Q and the prior P which, in our case, gives

$$\mathrm{KL}(Q_{\mathbf{i}} \| P) = \mathop{\mathbf{E}}_{\rho \sim Q_{\mathbf{i}}} \ln \frac{Q_{\mathbf{i}}(\boldsymbol{\rho})}{P_{\mathcal{I}}(\mathbf{i}) P_{\mathbf{i}}(\boldsymbol{\rho})}$$

For these classes of posteriors Q and priors P, the PAC-Bayes theorem of Laviolette and Marchand (2007, Corollary 8) reduces to the following simpler version.

Theorem 2 (Laviolette and Marchand 2007) *Given all our previous definitions, for any prior P and for any $\delta \in (0, 1]$*

$$\mathbf{P}_{\mathbf{Z}^m}\left(\forall Q_{\mathbf{i}} : \mathrm{kl}(R_S(G_{Q_{\mathbf{i}}}) \| R(G_{Q_{\mathbf{i}}})) \leq \frac{1}{m - |\mathbf{i}|}\left[\mathrm{KL}(Q_{\mathbf{i}} \| P) + \ln \frac{m+1}{\delta}\right]\right) \geq 1 - \delta$$

where

$$\mathrm{kl}(q \| p) \stackrel{\mathrm{def}}{=} q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p} \quad \textit{for } q < p$$

To obtain a bound for $R(G_{Q_{\mathbf{i}}})$ we need to specify $Q_{\mathbf{i}}(\boldsymbol{\rho})$, $P_{\mathcal{I}}(\mathbf{i})$, and $P_{\mathbf{i}}(\boldsymbol{\rho})$.

Since all vectors $\mathbf{i}$ having the same size $|\mathbf{i}|$ are, *a priori*, equally "good", we choose

$$P_{\mathcal{I}}(\mathbf{i}) = \frac{1}{\binom{m}{|\mathbf{i}|}} p(|\mathbf{i}|)$$

for any $p(\cdot)$ such that $\sum_{d=0}^{m} p(d) = 1$. As we saw earlier, in Sect. 4.2, it is generally preferable to choose, for $p(d)$, a slowly decreasing function of d since the risk bound will deteriorate for large $|\mathbf{i}|$.

For the specification of $P_{\mathbf{i}}(\boldsymbol{\rho})$, we assume that each radius value, in some predefined interval[8] $[0, R]$, is equally likely to be chosen for each ρ_i such that $i \in \mathbf{i}$. That is, we consider, for each ρ_i, $P(\rho_i) = \frac{1}{R}$ if $0 < \rho \leq R$ and zero otherwise. Here R is some "large" distance specified *a priori*. For $Q_{\mathbf{i}}(\boldsymbol{\rho})$, a margin interval $[a_i, b_i] \subseteq [0, R]$ of equally good radius values is chosen by the learner for each $i \in \mathbf{i}$. Hence, we choose

$$P_{\mathbf{i}}(\boldsymbol{\rho}) = \begin{cases} \prod_{i \in \mathbf{i}} \frac{1}{R} \; = \; \left(\frac{1}{R}\right)^{|\mathbf{i}|} & \text{if } 0 < \rho_i < R, \forall i \in \mathbf{i} \\ 0 & \text{otherwise} \end{cases}$$

and

$$Q_{\mathbf{i}}(\boldsymbol{\rho}) = \begin{cases} \prod_{i \in \mathbf{i}} \frac{1}{b_i - a_i} & \text{if } a_i < \rho_i < b_i, \forall i \in \mathbf{i} \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the Gibbs classifier returned by the learner will draw each radius ρ_i uniformly in $[a_i, b_i]$. A deterministic classifier is then specified by fixing each radius values $\rho_i \in [a_i, b_i]$. It is tempting at this point to choose $\rho_i = (a_i + b_i)/2$ $\forall i \in \mathbf{i}$ (i.e., in the middle

[8] Note that this quantity is *not* the same as the scale in Sect. 4.2.

of each interval). However, we will see shortly that the PAC-Bayes theorem offers a better guarantee for another type of deterministic classifier.

Consequently, with these choices for $Q_{\mathbf{i}}(\boldsymbol{\rho})$, $P_{\mathcal{I}}(\mathbf{i})$, and $P_{\mathbf{i}}(\boldsymbol{\rho})$, the KL divergence between $Q_{\mathbf{i}}$ and P is given by

$$KL(Q_{\mathbf{i}}\|P) = \ln\binom{m}{|\mathbf{i}|} + \ln\left(\frac{1}{p(|\mathbf{i}|)}\right) + \sum_{i\in\mathbf{i}} \ln\left(\frac{R}{b_i - a_i}\right)$$

Notice that the KL divergence is small for small values of $|\mathbf{i}|$ (whenever $p(|\mathbf{i}|)$ is not too small) and for large margin values $(b_i - a_i)$. Hence, the KL divergence term in Theorem 2 favors both sparsity (small $|\mathbf{i}|$) and large margins. Hence, in practice, the minimum might occur for some $G_{Q_{\mathbf{i}}}$ that sacrifices sparsity whenever larger margins can be found.

Since the posterior Q is identified by $\mathbf{i}$ and by the intervals $[a_i, b_i]$ $\forall i \in \mathbf{i}$, we will now refer to the Gibbs classifier $G_{Q_{\mathbf{i}}}$ by $G^{\mathbf{i}}_{\mathbf{ab}}$ where $\mathbf{a}$ and $\mathbf{b}$ are the vectors formed by the unions of a_is and b_is respectively. To obtain a risk bound for $G^{\mathbf{i}}_{\mathbf{ab}}$, we need to find a closed-form expression for $R_S(G^{\mathbf{i}}_{\mathbf{ab}})$. For this task, let $U[a,b]$ denote the uniform distribution over $[a,b]$ and let $\sigma^i_{a,b}(\mathbf{x})$ be the probability that a ball with center $\mathbf{x}_i$ assigns to $\mathbf{x}$ the class label y_i when its radius ρ is drawn according to $U[a,b]$:

$$\sigma^i_{a,b}(\mathbf{x}) \overset{\text{def}}{=} \Pr_{\rho\sim U[a,b]}(h_{i,\rho}(\mathbf{x}) = y_i) = \begin{cases} 1 & \text{if } d(\mathbf{x},\mathbf{x}_i) \le a \\ \frac{b-d(\mathbf{x},\mathbf{x}_i)}{b-a} & \text{if } a \le d(\mathbf{x},\mathbf{x}_i) \le b \\ 0 & \text{if } d(\mathbf{x},\mathbf{x}_i) \ge b \end{cases}$$

Therefore,

$$\xi^i_{a,b}(\mathbf{x}) \overset{\text{def}}{=} \Pr_{\rho\sim U[a,b]}(h_{i,\rho}(\mathbf{x}) = 1) = \begin{cases} \sigma^i_{a,b}(\mathbf{x}) & \text{if } y_i = 1 \\ 1 - \sigma^i_{a,b}(\mathbf{x}) & \text{if } y_i = 0 \end{cases}$$

Now let $G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x})$ denote the probability that the conjunction of balls outputs 1 when each $\rho_i \in \boldsymbol{\rho}$ are drawn according to $U[a_i, b_i]$. We then have

$$G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}) = \prod_{i\in\mathbf{i}} \xi^i_{a_i,b_i}(\mathbf{x})$$

Consequently, the risk $R_{(\mathbf{x},y)}(G^{\mathbf{i}}_{\mathbf{ab}})$ on a single example $(\mathbf{x}, y)$ is given by $G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x})$ if $y = 0$ and by $1 - G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x})$ otherwise. Therefore

$$R_{(\mathbf{x},y)}\big(G^{\mathbf{i}}_{\mathbf{ab}}\big) = y\big(1 - G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x})\big) + (1-y)G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}) = (1-2y)\big(G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}) - y\big)$$

Hence, the empirical risk $R_S(G^{\mathbf{i}}_{\mathbf{ab}})$ of the Gibbs classifier $G^{\mathbf{i}}_{\mathbf{ab}}$ is given by

$$R_S\big(G^{\mathbf{i}}_{\mathbf{ab}}\big) = \frac{1}{m - |\mathbf{i}|} \sum_{j\in\bar{\mathbf{i}}} (1-2y_j)\big(G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}_j) - y_j\big)$$

From this expression we see that $R_S(G^{\mathbf{i}}_{\mathbf{ab}})$ is small when $G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}_j)$ is close to y_j $\forall j \in \bar{\mathbf{i}}$. Training points where $G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}_j) \approx 1/2$ should therefore be avoided.

The PAC-Bayes theorem below provides a risk bound for the Gibbs classifier $G^{\mathbf{i}}_{\mathbf{ab}}$. Since the Bayes classifier $B^{\mathbf{i}}_{\mathbf{ab}}$ just performs a majority vote under the same posterior distribution as the one used by $G^{\mathbf{i}}_{\mathbf{ab}}$, we have that $B^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}) = 1$ iff $G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}) > 1/2$. From the above definitions, note that the decision surface of the Bayes classifier, given by $G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}) = 1/2$, differs from the decision surface of a deterministic classifier in which the boundary for each feature is fixed at the center of each interval, i.e., $\rho_i = (a_i + b_i)/2\ \forall i \in \mathbf{i}$. Let us denote such a deterministic classifier by $C_{\mathbf{i}\boldsymbol{\rho}}$. In fact there does not exists any classifier $C_{\mathbf{i}\boldsymbol{\rho}}$ that has the same decision surface as the Bayes classifier $B^{\mathbf{i}}_{\mathbf{ab}}$. From the relation between $B^{\mathbf{i}}_{\mathbf{ab}}$ and $G^{\mathbf{i}}_{\mathbf{ab}}$, it also follows that $R_{(\mathbf{x},y)}(B^{\mathbf{i}}_{\mathbf{ab}}) \le 2R_{(\mathbf{x},y)}(G^{\mathbf{i}}_{\mathbf{ab}})$ for any $(\mathbf{x}, y)$. Consequently, $R(B^{\mathbf{i}}_{\mathbf{ab}}) \le 2R(G^{\mathbf{i}}_{\mathbf{ab}})$. Hence, we have the following theorem.

Corollary 4 *Given all our previous definitions, for any* $\delta \in (0, 1]$, *for any* p *satisfying* $\sum_{d=0}^{m} p(d) = 1$, *and for any fixed distance value* R, *we have*:

$$\mathbf{P}_{\mathbf{Z}^m}\Bigg(\forall \mathbf{i}, \mathbf{a}, \mathbf{b}\colon\ R\big(G^{\mathbf{i}}_{\mathbf{ab}}\big) \le \sup\Bigg\{\epsilon\colon \mathrm{kl}\big(R_S\big(G^{\mathbf{i}}_{\mathbf{ab}}\big)\|\epsilon\big) \le \frac{1}{m - |\mathbf{i}|}\Bigg[\ln\binom{m}{|\mathbf{i}|} + \ln\bigg(\frac{1}{p(|\mathbf{i}|)}\bigg) + \sum_{i\in\mathbf{i}} \ln\bigg(\frac{R}{b_i - a_i}\bigg) + \ln\frac{m+1}{\delta}\Bigg]\Bigg\}\Bigg) \ge 1 - \delta$$

Furthermore: $R(B^{\mathbf{i}}_{\mathbf{ab}}) \le 2R(G^{\mathbf{i}}_{\mathbf{ab}})\ \forall \mathbf{i}, \mathbf{a}, \mathbf{b}$.

Recall that the KL divergence is small for small values of $|\mathbf{i}|$ (whenever $p(|\mathbf{i}|)$ is not too small) and for large margin values $(b_i - a_i)$. Furthermore, the Gibbs empirical risk $R_S(G^{\mathbf{i}}_{\mathbf{ab}})$ is small when the training points are located far away from the Bayes decision surface $G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}) = 1/2$ (with $G^{\mathbf{i}}_{\mathbf{ab}}(\mathbf{x}_j) \to y_j\ \forall j \in \bar{\mathbf{i}}$). *Consequently, the Gibbs classifier with the smallest guarantee of risk should perform a non trivial margin-sparsity tradeoff.*

7 Learning in the premise of Corollary 4: a soft greedy approach

Corollary 4 suggests that the learner should try to find the Bayes classifier $B^{\mathbf{i}}_{\mathbf{ab}}$ that uses a small number of balls (i.e., a small $|\mathbf{i}|$), each with a large separating margin $(b_i - a_i)$, while keeping the empirical Gibbs risk $R_S(G^{\mathbf{i}}_{\mathbf{ab}})$ at a low value. To achieve this goal, we have adapted the greedy algorithm for the set covering machine (SCM).

In our case, however, we need to keep the Gibbs risk on S low instead of the risk of a deterministic classifier. Since the Gibbs risk is a "soft measure" that uses the piece-wise linear functions $\sigma^i_{a,b}$ instead of "hard" indicator functions, we need a "softer" version of the utility function U_i. Indeed, a negative example that falls in the linear region of a $\sigma^i_{a,b}$ is in fact partly covered. Following this observation, let $\mathbf{k}$ be the vector of indices of the examples that we have used as ball centers so far for the construction of the classifier. Let us first define the *covering value* $\mathcal{C}(G^{\mathbf{k}}_{\mathbf{ab}})$ of $G^{\mathbf{k}}_{\mathbf{ab}}$ by the "amount" of negative examples assigned to class 0 by $G^{\mathbf{k}}_{\mathbf{ab}}$:

$$\mathcal{C}\big(G^{\mathbf{k}}_{\mathbf{ab}}\big) \overset{\text{def}}{=} \sum_{j\in\bar{\mathbf{k}}} (1 - y_j)\big[1 - G^{\mathbf{k}}_{\mathbf{ab}}(\mathbf{x}_j)\big]$$

We also define the *positive-side error* $\mathcal{E}(G^{\mathbf{k}}_{\mathbf{ab}})$ of $G^{\mathbf{k}}_{\mathbf{ab}}$ as the "amount" of positive examples assigned to class 0:

$$\mathcal{E}\big(G^{\mathbf{k}}_{\mathbf{ab}}\big) \overset{\text{def}}{=} \sum_{j\in\bar{\mathbf{k}}} y_j\big[1 - G^{\mathbf{k}}_{\mathbf{ab}}(\mathbf{x}_j)\big]$$

We now want to add another ball, centered on an example with index i, to obtain a new vector $\mathbf{k}'$ containing this new index in addition to those present in $\mathbf{k}$. Hence, we now introduce the *covering contribution* of ball i (centered on $\mathbf{x}_i$) as

$$\begin{aligned}\mathcal{C}^{\mathbf{k}}_{\mathbf{ab}}(i) &\overset{\text{def}}{=} \mathcal{C}\big(G^{\mathbf{k}'}_{\mathbf{a}'\mathbf{b}'}\big) - \mathcal{C}\big(G^{\mathbf{k}}_{\mathbf{ab}}\big)\\ &= (1-y_i)\big[1-\xi^i_{a_i,b_i}(\mathbf{x}_i)\,G^{\mathbf{k}}_{\mathbf{ab}}(\mathbf{x}_i)\big]\\ &\quad + \sum_{j\in\overline{\mathbf{k}'}}(1-y_j)\big[1-\xi^i_{a_i,b_i}(\mathbf{x}_j)\big]G^{\mathbf{k}}_{\mathbf{ab}}(\mathbf{x}_j)\end{aligned}$$

and the *positive-side error contribution* of ball i as

$$\begin{aligned}\mathcal{E}^{\mathbf{k}}_{\mathbf{ab}}(i) &\overset{\text{def}}{=} \mathcal{E}\big(G^{\mathbf{k}'}_{\mathbf{a}'\mathbf{b}'}\big) - \mathcal{E}\big(G^{\mathbf{k}}_{\mathbf{ab}}\big)\\ &= y_i\big[1-\xi^i_{a_i,b_i}(\mathbf{x}_i)\,G^{\mathbf{k}}_{\mathbf{ab}}(\mathbf{x}_i)\big] + \sum_{j\in\overline{\mathbf{k}'}} y_j\big[1-\xi^i_{a_i,b_i}(\mathbf{x}_j)\big]G^{\mathbf{k}}_{\mathbf{ab}}(\mathbf{x}_j)\end{aligned}$$

Typically, the covering contribution of ball i should increase its "utility" and its positive-side error should decrease it. Hence, we define the *utility* $U^{\mathbf{k}}_{\mathbf{ab}}(i)$ *of adding ball i to* $G^{\mathbf{k}}_{\mathbf{ab}}$ as

$$U^{\mathbf{k}}_{\mathbf{ab}}(i) \overset{\text{def}}{=} \mathcal{C}^{\mathbf{k}}_{\mathbf{ab}}(i) - p\cdot\mathcal{E}^{\mathbf{k}}_{\mathbf{ab}}(i)$$

where parameter p represents the *penalty* of misclassifying a positive example. For a fixed value of p, the "soft greedy" algorithm simply consists of adding, to the current Gibbs classifier, a ball with maximum added utility until either the maximum number of possible features (balls) has been reached or that all the negative examples have been (totally) covered. It is understood that, during this soft greedy algorithm, we can remove an example $(\mathbf{x}_j, y_j)$ from S whenever it is totally covered. This occurs whenever $G^{\mathbf{k}}_{\mathbf{ab}}(\mathbf{x}_j)=0$.

The term $\sum_{i\in\mathbf{i}}\ln(R/(b_i-a_i))$, present in the risk bound of Corollary 4, favors "soft balls" having large margins $b_i - a_i$. Hence, we introduce a *margin parameter* $\gamma \geq 0$ that we use as follows. At each greedy step, we first search among balls having $b_i - a_i = \gamma$. Once such a ball, of center $\mathbf{x}_i$, having maximum utility has been found, we try to increase further its utility by searching among all possible values of a_i and $b_i > a_i$ while keeping its center $\mathbf{x}_i$ fixed.[9] Both p and γ will be chosen by cross validation on the training set.

8 Empirical results on natural data

We have compared the new information theoretic learning algorithm (called here SCM-IT), that codes each ball radius with a message string, and the new PAC-Bayes learning algorithm (called here SCM-PB) with the old algorithm (called here SCM), that codes each radius with a training example. All of these algorithms were also compared with the support vector machine (SVM) equipped with a RBF kernel of variance γ 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 11 UCI data sets. Each data set was randomly split in two parts. About half of the examples were used for training and the remaining examples were

[9] The possible values for a_i and b_i are defined by the location of the training points.

Table 1 Results of SVM and classical SCM on UCI datasets

Data set			SVM results				SCM-cv		SCM-b	
Name	Train	Test	C	γ	SVs	Errs	b	Errs	b	Errs
Breastw	343	340	1	5	38	0.044	2	**0.032**	1	0.035
Bupa	170	175	2	0.17	169	**0.377**	2	0.405	2	0.4
Credit	353	300	100	2	282	**0.17**	12	0.216	1	0.19
Glass	107	107	10	0.17	51	0.271	4	0.186	4	**0.177**
Haberman	144	150	2	1	81	**0.26**	2	0.273	1	**0.26**
Heart	150	147	1	0.17	64	0.176	1	0.190	1	**0.156**
USvotes	235	200	1	25	53	**0.065**	8	0.13	3	0.095
Diabetis	468	300	1	5	432	0.413	5	**0.263**	5	**0.263**
German	500	500	100	1.50	498	0.446	36	0.288	10	**0.286**
Thyroid	107	108	5	5	88	0.398	2	**0.074**	1	0.083
Ionospher	175	176	1	1.50	153	0.090	7	**0.045**	7	**0.045**

Table 2 SCM-IT and PAC-Bayes-SCM results on UCI datasets

Data set			SCM-IT-cv			SCM-IT-b			SCM-PB-cv			SCM-PB-b		
Name	Train	Test	b	l^*	Errs	b	l^*	Errs	b	γ	Errs	b	γ	Errs
Breastw	343	340	1	3	0.035	1	1	0.035	4	0.08	**0.029**	1	0.05	**0.029**
Bupa	170	175	2	7	0.394	11	7	**0.382**	6	0.1	**0.382**	1	0.05	0.417
Credit	353	300	11	6	0.163	8	5	**0.153**	11	0.09	0.183	2	0.03	0.21
Glass	107	107	7	6	0.177	3	5	**0.168**	16	0.04	0.177	4	0.02	0.233
Haberman	144	150	8	2	**0.24**	2	2	0.246	1	0.2	0.253	1	0.12	**0.24**
Heart	150	147	1	2	0.163	1	2	**0.156**	1	0	0.190	1	0.04	0.163
USvotes	235	200	7	3	0.095	4	2	0.075	18	0.14	**0.06**	1	0.03	0.09
Diabetis	468	300	12	7	0.266	88	0	0.31	2	0.02	0.27	1	0.18	**0.266**
German	500	500	19	7	0.294	3	3	0.286	43	0.16	**0.274**	1	0.08	0.288
Thyroid	107	108	3	0	0.064	3	0	**0.046**	3	0.04	0.055	1	0.06	0.064
Ionospher	175	176	5	3	0.039	9	0	0.039	10	0.03	**0.011**	5	0.03	0.039

used for testing. The corresponding values for these numbers of examples are given in the "train" and "test" columns of Tables 1 and 2. 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 split) was used to determine the learning parameters of SCM, SCM-IT and SCM-PB. These results are referred to (in Tables 1 and 2) as SCM-cv, SCM-IT-cv and SCM-PB-cv.

The "SVs" column of the SVM results refers to the number of support vectors present in the final classifier. The "errs" column, for all learning algorithms, refers to the fraction of classification errors obtained on the testing set. Finally, the "b", "l^*" and "γ" columns of the SCM results refer, respectively, to the number of balls, the maximum number of bits used by the final classifier in the case of SCM-IT and the margin parameter (divided by the average distance between the positive and the negative examples) in the case of SCM-PB. The results reported for SCM-PB (in Table 2) refer to the Bayes classifier only. The results for the Gibbs classifier are similar. Note, however, that in the case of SCM-PB the algorithm is optimized for the Gibbs classifier. The reported result for the SCM-PB Bayes classifier are the ones for which the corresponding Gibbs risk is optimized. Finally, the best error rates obtained over all the approaches are presented in bold face.

8.1 Learning by bound minimization

A risk bound defines an optimization problem for learning algorithms: the algorithm should find the classifier with the smallest risk bound. We see here how our risk bound performs in practice to select the best classifier from the same classifier space as in the case of above mentioned CV experiments. That is, we have the *same* possible choices of the learning parameters that we have used for the 5-fold CV method.[10] For each of the proposed approaches of Sects. 4.2 and 6, we use the risk bounds of Corollary 3 and Corollary 4 to perform the model selection and therefore output the classifier that minimizes the respective bounds. We test the classifier thus chosen on the testing set and report the results. For the original SCM, we use the bound of Corollary 2. These results are referred to (in Tables 1 and 2) as SCM-b, SCM-IT-b and SCM-PB-b respectively. The aim of this evaluation is to assess whether explicit bound minimization can indeed be competitive with cross-validation.

Finally, Fig. 1 compares the performance of the various versions of the SCM to that of the SVM when performing the model selection from cross validation (Fig. 1(a)) as well as compares the performance of the classical SCM, SCM-IT and SCM-PB when the risk bounds guides the model selection process as opposed to cross validation for each method against an SVM benchmark (Figs. 1(b), 1(c) and 1(d), respectively).

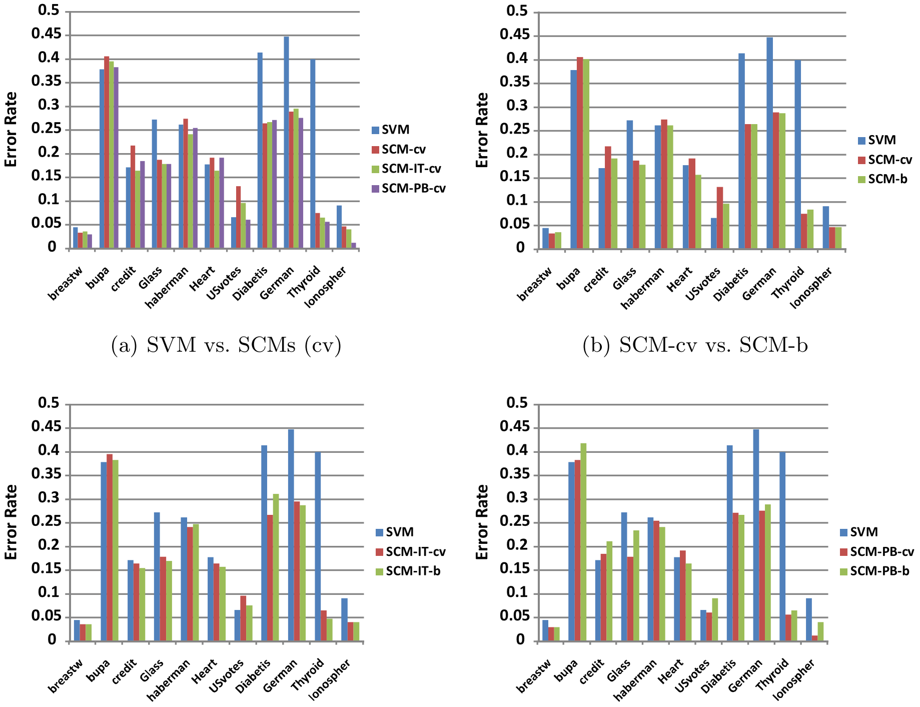
9 Analysis and discussion: putting the pieces together

The risk bounds based on dispensing the reconstruction information between the compression set and the additional information efficiently has a crucial balancing effect. This also offers an explicit control over the margin and sparsity of the classifier. Moreover, these risk bounds are pragmatic and can successfully guide the model selection process. This is quite important since they afford new optimization criteria for the learning algorithm in addition to empirical risk minimization. This also demonstrates the effectiveness of the sample compression framework in offering tight and pragmatic bounds usable in practice. Moreover, the bounds in this framework are general enough to be applied to a variety of learning algorithms and settings. See for instance, the work of Shah (2007) for a similar approach applied in the context of decision tree learning algorithm and Shah (2006) in the context of learning conjunctions of decision stumps applied to microarray data. We now analyze the main observations of this work.

9.1 Risk bounds

An efficient redistribution of reconstruction information resulted not only in tighter bounds but also alternate learning strategies for the SCM. The bounds of Corollary 3 and Corollary 4

[10] It consists of an *exhaustive* list of possible values for (p, l^*, v).



(a) SVM vs. SCMs (cv)

(b) SCM-cv vs. SCM-b

(c) SCM-IT-cv vs. SCM-IT-b

(d) SCM-PB-cv vs. SCM-PB-b

Fig. 1 Error rates comparison for the SVM, the classical SCM, SCM-IT and SCM-PB. The suffixes -cv and -b denote the test error rates for respective SCM when the model selection was performed using cross-validation and the risk bound respectively. Part (**a**) compares the SVM, SCM, SCM-IT and SCM-PB all with cross validation based model selection. Parts (**b**), (**c**) and (**d**) compare the algorithm performance of, respectively, SCM, SCM-IT and SCM-PB when the risk bounds are used for model selection to that of cross validation-based model selection. In each case, results of the SVM are shown as benchmarks

are tighter than the bounds proposed for the SCM earlier in Marchand and Shawe-Taylor (2001) and Marchand and Shawe-Taylor (2002). They are also tighter than the binomial tail inversion version of the bound for the SCM with data-dependent balls in the original setting (using a training example to signify the border) in Corollary 2. In the new versions, the compression set now consists of only one example per ball unlike the previous versions while the radius information is contained in the message distribution.

Moreover, the new strategy also enables an explicit control over trading off sparsity in favor of a larger magnitude of the separating margin. We examine this effect in more detail next.

9.2 Empirical results and margin-sparsity trade-off

Let us analyze the empirical results with regard to some criteria of interest starting with the test performance of the learning algorithms resulting from the new formulations of Sects. 5 and 7. We discuss, in this subsection, the results when the model selection is done using cross validation. Let us analyze these for both the new formulations of the SCM. Note that the results of Table 2 should be compared to the results for the original formulation of SCM presented in Table 1.

For the information theoretic formulation "SCM-IT", the margin-sparsity trade-off effect is notably visible in the case of Credit, Haberman and USVotes datasets and to some extent in the case of Thyroid dataset (see the "errs" column under "SCM-IT-cv" in Table 2). The case of credit and USVotes dataset are the cases when the algorithm opts for sparser classifiers at the cost of higher margins (see the l^* column under "SCM-IT-cv" in Table 2). In contrast, the learning algorithm sacrifices some sparsity in favor of margin in the case of Haberman dataset to achieve better accuracy.

In the case of the PAC-Bayes SCM formulation "SCM-PB", the margin-sparsity effect is prominent in the case of USVotes, German and Ionosphere datasets and to a lesser extent in the case of Bupa and Credit datasets (see the "errs" column under "SCM-PB-cv" in Table 2). In the case of the USVotes, German and Ionosphere, the algorithm sacrifices sparsity significantly in favor of higher margins (see the "γ" column under "SCM-PB-cv" in Table 2). A similar effect with marginal gain in accuracy can be seen in the case of bupa dataset as well. In the case of credit, the algorithm favor a relatively sparse solution as compared to the SCM's original formulation.

For other datasets, in both the SCM-IT and SCM-PB cases, the improvements are generally marginal as compared to the original SCM formulation. However, when compared to the SVM (Table 1), both the original SCM and the new formulations consistently perform better except in the case of bupa where SVM gives marginally better results.

9.3 Learning by bound minimization

Let us analyze the effectiveness of the proposed risk bounds of Sects. 4.2 and 6 in guiding the model selection process. The information theoretic bound of Sect. 4.2 appears to perform better than the PAC-Bayesian formulation in general. The models selected by bound in the case of SCM-IT (see the columns under "SCM-IT-b" in Table 2) are generally comparable and in some cases better than the ones selected using cross-validation. The most notable results are in the case of credit, USVotes and Thyroid datasets where the bound yields better results than the cross-validation. In fact, the results for the Thyroid dataset using the model-selection via bound are the best over all the approaches.

In contrast, the PAC-Bayes bound of Sect. 6, although tight, is not as effective in performing model selection as the information theoretic bound. Nevertheless, the results obtained via model selection using this bound (see the columns under "SCM-PB-b" in Table 2) are still comparable to the ones obtained by cross-validation.

The main reason for this difference in performance seems to be the importance given to sparse classifiers by the PAC-Bayes bound. This can be easily seen in the empirical results (see column "b" under "SCM-PB-b" in Table 2) where the models selected by the bound are consistently sparser (and mostly with very small margins as seen in the "γ" column under "SCM-PB-b" in Table 2) than those selected via cross-validation.

9.4 Time complexity analysis

For SCM with data-dependent balls, the algorithm's time complexity is shown to be $O(m^2 \log(m))$ (Marchand and Shawe-Taylor 2002, Theorem 9) where m is the number of training examples. In the information theoretic approach of Sect. 4.2, we maintain the original complexity of $O(m^2 \log(m))$ for the SCM with data-dependent balls.[11] However, in the

[11] Although, we have a slight advantage in that we do not test over the boundaries of each example. However, this does not change the worst case complexity.

PAC-Bayes formulation of Sect. 7 of the SCM algorithm, we have about an extra $\log(m)$ factor in overall time complexity as described below.

9.4.1 Time complexity analysis for the PAC-Bayes formulation

We analyze the running time of PAC-Bayes soft greedy SCM learning algorithm of Sect. 6 for fixed p and γ. For each potential ball center, we first sort the $m-1$ other examples with respect to their distances from the center in $O(m\log m)$ time. Then, for this center $\mathbf{x}_i$, the set of a_i values that we examine are those specified by the distances (from $\mathbf{x}_i$) of the $m-1$ sorted examples.[12] Since the examples are sorted, it takes time $\in O(km)$ to compute the covering contributions and the positive-side error *for all* the $m-1$ values of a_i. Here k is the largest number of examples falling into the margin. We are always using small enough γ values to have $k \in O(\log m)$ since, otherwise, the results are terrible. It therefore takes time $\in O(m\log m)$ to compute the utility values of all the $m-1$ different balls of a given center. This gives a time $\in O(m^2\log m)$ to compute the utilities for all the possible m centers. Once a ball with a largest utility value has been chosen, we then try to increase further its utility by searching among $O(m^2)$ pair values for (a_i, b_i). We then remove the examples covered by this ball and repeat the algorithm on the remaining examples. It is well known that greedy algorithms of this kind have the following guarantee: if there exist r balls that cover all the m examples, the greedy algorithm will find at most $r\ln(m)$ balls. Since we almost always have $r \in O(1)$, the running time of the whole algorithm will almost always be $\in O(m^2\log^2(m))$.

10 Conclusion and future work

In this work, we investigated the classifiers in the sample compression framework that are specified by two distinct sources of information: a *compression set* and a *message string* of additional information. In the compression setting, a reconstruction function specifies a classifier when given this information. We examined how an efficient redistribution of reconstruction information can lead to more general classifiers. More particularly, we showed how we can obtain risk bounds that can provide an explicit control over these two quantities: the classifier sparsity and the magnitude of its separating margin. We further showed how a non-trivial margin-sparsity trade-off can be achieved by such redistribution; and that when such a trade-off is achieved in practice, classifiers with better performance can be obtained.

In this paper, we mostly worked with the set covering machine algorithm with data-dependent balls. However, the generic risk bounds proposed in Sects. 4.2 and 6 are general enough to be applied to other learning settings. (See for instance, Shah 2007; Shah 2006 for application on decision trees and conjunctions of decision stumps respectively.)

Another very important issue that we address in this work is the practical utility of the proposed risk bounds in what is termed as *learning by bound minimization*. This provides an important alternative to other learning strategies, esp. empirical and structural risk minimization. We showed, how the proposed bound can guide the model selection process and yield at least as good as (or sometimes better) classifier when compared to traditional cross validation based model selection approach. One of the main advantages of such sample compression based risk bounds over other frameworks comes from the independence of

[12] Recall that for each value of a_i, the value of b_i is set to $a_i+\gamma$ at this stage.

risk bounds from (explicit inclusion of) hypothesis class complexity considerations. This property not only makes the proposed bounds quite useful in practice but also makes them applicable to data-dependent learning settings. This is in contrast with hypothesis class complexity dependent bounds such as VC or Rademacher complexity-based bounds. Empirical results validate these observations.

An interesting dimension to investigate in the future would be the extension of this approach to the case of SCM with half-spaces (Marchand et al. 2003).

Similarly, other classes of decision based learners such as decision lists can also yield interesting results. However, the approaches proposed here should be viewed in their general applicability to classifiers in sample compression settings where the reconstruction information can be dispensed efficiently between compression sets and messages so as to obtain explicit control over the sparsity and the magnitude of the separating margins of the classifiers.

Acknowledgements M.M. and F.L. would like to acknowledge the support of National Science and Engineering Research Council (NSERC) of Canada through Discovery Grant Nos. 122405 and 262067 respectively. The authors would like to thank the action editor and the anonymous reviewers for their comments and suggestions that helped to improve the manuscript significantly.

Appendix A: Proof of Lemma 1

We first show that

$$\mathrm{Bin}(k,m,r) \stackrel{\mathrm{def}}{=} \sum_{i=0}^{k} \binom{m}{i} r^i (1-r)^{m-i} \leq \binom{m}{k}(1-r)^{m-k}$$

Let f be a classifier with risk $R(f) = r$. Recall that the binomial tail distribution $\mathrm{Bin}(k,m,r)$ 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:

$$\begin{aligned}
\mathrm{Bin}(k,m,r) &\stackrel{\mathrm{def}}{=} \sum_{i=0}^{k} \binom{m}{i} r^i (1-r)^{m-i} \\
&= \Pr\{\exists S \subseteq \{1,2,\ldots,m\} \wedge |S| = m-k : R_S(f) = 0\} \\
&\leq \sum_{S\subseteq\{1,\ldots,m\}:\ |S|=m-k} \Pr\{R_S(f)=0\} \quad \text{(the union bound)} \\
&= \binom{m}{m-k}(1-r)^{m-k} = \binom{m}{k}(1-r)^{m-k} \\
&\stackrel{\mathrm{def}}{=} g(k,m,r)
\end{aligned}$$

Hence, the tail of the binomial is a decreasing function of r when k and m are fixed. It follows that:

$$\begin{aligned}
\overline{\mathrm{Bin}}(k,m,\delta) &\stackrel{\mathrm{def}}{=} \sup\{r : \mathrm{Bin}(k,m,r) \geq \delta\} \\
&\leq \sup\{r :\ g(k,m,r) \geq \delta\} \\
&= \{r : g(k,m,r) = \delta\}
\end{aligned}$$

The value of r that satisfies the equation $g(k,m,r)=\delta$ is precisely given by:

$$r \;=\; 1-\exp\left[-\frac{1}{m-k}\left(\ln\binom{m}{k}+\ln\frac{1}{\delta}\right)\right]$$

Hence,

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

References

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

Blum, A., & Langford, J. (2003). PAC-MDL bounds. In *Lecture notes in artificial intelligence: Vol. 2777. Proceedings of 16th annual conference on learning theory, COLT 2003*, Washington, DC, August 2003 (pp. 344–357). Berlin: Springer.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. (1987). Occam's razor. *Information Processing Letters*, *24*, 377–380.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual ACM workshop on computational learning theory* (pp. 144–152). New York: ACM.

Chvátal, V. (1979). A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, *4*, 233–235.

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

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability, a guide to the theory of NP-completeness*. New York: Freeman.

Graepel, T., Herbrich, R., & Shawe-Taylor, J. (2000). Generalisation error bounds for sparse linear classifiers. In *Proceedings of the thirteenth annual conference on computational learning theory* (pp. 298–303).

Graepel, T., Herbrich, R., & Shawe-Taylor, J. (2005). PAC-Bayesian compression bounds on the prediction error of learning algorithms for classification. *Machine Learning*, *59*(12), 55–76.

Graepel, T., Herbrich, R., & Williamson, R. C. (2001). From margin to sparsity. In *Advances in neural information processing systems* (Vol. 13, pp. 210–216).

Haussler, D. (1988). Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, *36*, 177–221.

Kearns, M. J., & Vazirani, U. V. (1994). *An introduction to computational learning theory*. Cambridge: MIT Press.

Kuzmin, D., & Warmuth, M. K. (2007). Unlabeled compression schemes for maximum classes. *Journal of Machine Learning Research*, *8*, 2047–2081.

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

Laviolette, F., & Marchand, M. (2007). PAC-Bayes risk bounds for stochastic averages and majority votes of sample-compressed classifiers. *Journal of Machine Learning Research*, *8*, 1461–1487.

Laviolette, F., Marchand, M., & Shah, M. (2005). Margin-sparsity trade-off for the set covering machine. In *Lecture notes in artificial intelligence: Vol. 3720. Proceedings of the 16th European conference on machine learning, ECML 2005* (pp. 206–217). Berlin: Springer.

Laviolette, F., Marchand, M., & Shah, M. (2006). A PAC-Bayes approach to the set covering machine. In *Advances in neural information processing systems* (Vol. *18*, pp. 731–738). Cambridge: MIT Press.

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

Marchand, M., Shah, M., Shawe-Taylor, J., & Sokolova, M. (2003). The set covering machine with data-dependent half-spaces. In *Proceedings of the twentieth international conference on machine learning (ICML 2003)* (pp. 520–527).

Marchand, M., & Shawe-Taylor, J. (2001). Learning with the set covering machine. In *Proceedings of the eighteenth international conference on machine learning (ICML 2001)* (pp. 345–352).

Marchand, M., & Shawe-Taylor, J. (2002). The set covering machine. *Journal of Machine Learning Reasearch*, *3*, 723–746.

Marchand, M., & Sokolova, M. (2005). Learning with decision lists of data-dependent features. *Journal of Machine Learning Reasearch*, *6*, 427–451.

McAllester, D. (2003). PAC-Bayesian stochastic model selection. *Machine Learning*, *51*, 5–21. A preliminary version appeared in proceedings of COLT'99.

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

Rubinstein, J. H., & Rubinstein, B. I. P. (2008). Geometric & topological representations of maximum classes with applications to sample compression. In *COLT* (pp. 299–310).

Shah, M. (2006). *Sample compression, margins and generalization: extensions to the set covering machine*. PhD thesis, SITE, University of Ottawa, Ottawa, Canada, May 2006.

Shah, M. (2007). Sample compression bounds for decision trees. In *ICML'07: Proceedings of the 24th international conference on machine learning* (pp. 799–806). New York: ACM.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the Association of Computing Machinery*, *27*(11), 1134–1142.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

von Luxburg, U., Bousquet, O., & Schölkopf, B. (2004). A compression approach to support vector model selection. *Journal of Machine Learning Research*, *5*, 293–323.

Warmuth, M. K. (2003). Compressing to VC dimension many points. In *Proceedings of the 16th annual conference on learning theory (COLT 03)* (pp. 743–744). Berlin: Springer.