Journal of Machine Learning Research 8 (2007) 2533-2549 Submitted 3/06; Revised 5/07; Published 11/07

Revised Loss Bounds for the Set Covering Machine and Sample-Compression Loss Bounds for Imbalanced Data

Zakria Hussain Z.HUSSAIN@CS.UCL.AC.UK
Centre for Computational Statistics and Machine Learning
University College London
London, UK, WC1E 6BT

François Laviolette FRANCOIS.LAVIOLETTE@IFT.ULAVAL.CA
Mario Marchand MARIO.MARCHAND@IFT.ULAVAL.CA
Départment IFT-GLO
Université Laval
Québec, Canada, G1V 0A6

John Shawe-Taylor JST@CS.UCL.AC.UK
Centre for Computational Statistics and Machine Learning
University College London
London, UK, WC1E 6BT

Spencer Charles Brubaker BRUBAKER@CC.GATECH.EDU
Matthew D. Mullin MDMULLIN@CC.GATECH.EDU
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, 30332

Editor: Gábor Lugosi

Abstract

Marchand and Shawe-Taylor (2002) have proposed a loss bound for the set covering machine that has the property to depend on the observed fraction of positive examples and on what the classifier achieves on the positive training examples. We show that this loss bound is incorrect. We then propose a loss bound, valid for any sample-compression learning algorithm (including the set covering machine), that depends on the observed fraction of positive examples and on what the classifier achieves on them. We also compare numerically the loss bound proposed in this paper with the incorrect bound, the original SCM bound and a recently proposed loss bound of Marchand and Sokolova (2005) (which does not depend on the observed fraction of positive examples) and show that the latter loss bounds can be substantially larger than the new bound in the presence of imbalanced misclassifications.

Keywords: set covering machines, sample-compression, loss bounds

1. Introduction

One of the key objectives of learning theory is to identify classes of functions and associated learning algorithms that deliver hypotheses with good guarantees of test set performance for a range of practical applications. Support vector machines (SVMs) have achieved this objective for problems for which classifiers with large margins can be identified. An alternative guiding principle for the selection of classifiers with guarantees of good generalization is to require some type of parsimony in the form of the functions. Typically seeking parsimonious solutions reduces to an NP-hard optimization, but an algorithm that delivers a good approximation to the optimal solution using a greedy approach is the so-called *set covering machine (SCM)*. This approach for producing very sparse classifiers having good generalization was proposed by Marchand and Shawe-Taylor (2001).

A *generalization error bound* is an upper bound on the expected test set performance that holds with high probability over the (random) choice of the training set. There are three ways in which such a bound can assist a user of adaptive systems technology. Firstly, the existence of such a bound justifying the form of a learning algorithm gives confidence in its reliability. This is the primary role of SVM bounds in most applications. The second is to guide model selection through setting regularization and hyperparameters to optimize the bound. The third is to give to users as a measure of performance. Experiments with recent SVM bounds have begun to make progress with the second goal, but it is fair to say that the third goal is still to be realized.

The situation with SCM bounds is that they are typically tighter and more reliable than those derived for SVMs. The classifier output by the SCM is described by a small subset of the training data called the *compression set*. By adapting the pioneering work of Littlestone and Warmuth (1986) on sample-compression schemes, Marchand and Shawe-Taylor (2001) have been able to obtain a loss bound that depends on the size (i.e., the number of examples) of the compression set of the SCM classifier. More recently, Marchand and Sokolova (2005) have been able to obtain a tighter bound, which applies to any sample-compression learning algorithm (including the SCM), by making use of sample-compression-dependent sets of messages. None of these loss bounds, however, depend on the observed fraction of positive examples in the training set and on the fraction of positive examples used for the compression set of the final classifier. Consequently, these loss bounds are not appropriate for identifying classifiers that perform well under frequently encountered distributions where the examples of one class are much more abundant than the examples of the other class (the class imbalance case) or when the loss suffered by misclassifying a positive example differs greatly from the loss suffered by misclassifying a negative example (the asymmetrical loss case).

To obtain a loss bound that reflects more accurately the performance of classifiers trained on imbalanced data sets, Marchand and Shawe-Taylor (2002) have proposed a SCM loss bound that depends on the observed fraction of positive examples in the training set and on the fraction of positive examples used for the compression set of the final classifier. However, we will show in Section 3 that this bound is *incorrect*. We then propose, in Section 4, a loss bound which is valid for any sample-compression learning algorithm (including the SCM) and that depends on the observed fraction of positive examples and on what the classifier achieves on the positive training examples. The proof of this new loss bound turns out to be much more involved than all other sample-compression loss bounds that do not depend on the observed fraction of positive examples (as in Marchand and Sokolova, 2005). Finally, for the SCM case, we compare numerically the loss bound of Section 4 with the recently proposed loss bound of Marchand and Sokolova (2005) (which does not depend on the observed fraction of positive examples) and show that the latter loss bound can be substantially larger than the former in the presence of imbalanced misclassifications.

The novelty of this paper is that we correct a bound that was found to be wrong, but in doing so, we derive a more general form that allows any learning algorithm, relying on sample compression schemes, to be upper bounded. Separately bounding the positive and negative errors also gives rise to a natural extension—namely asymmetric loss of sample compression risk bounds. In these cases, we give a higher weight for misclassification of one class over the other, typically because it is far less frequent than the dominant class. An example is classification of news articles by topic, where classification of all documents as *not* relevant will give good performance if we do not impose a greater cost on the misclassification of a relevant document.

We begin our discussion with preliminary definitions and terminology that will be used throughout the remainder of this paper.

2. Preliminary Definitions

Let the input space X be a set of n-dimensional vectors of $\mathbb{R}^n$ and let $\mathbf{x}$ be a member of X. We define a *feature* as an arbitrary Boolean-valued function that maps X onto $\{0,1\}$.

Consider any set $\mathcal{H} = \{h_i\}_{i=1}^{|\mathcal{H}|}$ of Boolean-valued features h_i. We will consider learning algorithms that are given any such set $\mathcal{H}$ and return a small subset $\mathcal{R} \subset \mathcal{H}$ of features. Given that subset $\mathcal{R}$, and an arbitrary input vector $\mathbf{x}$, the output $f(\mathbf{x})$ of the Set Covering Machine (SCM) is given by the conjunction

$$f(\mathbf{x}) \;=\; \bigwedge_{i\in\mathcal{R}} h_i(\mathbf{x})\,.$$

The function f contains a conjunction of features h_i that individually give outputs $h_i(\mathbf{x})$ of 0 or 1 to denote whether the input vector $\mathbf{x}$ belongs to class 0 or class 1, respectively. Therefore, the function f outputs 0 or 1 according to a conjunction of features h_i. A positive example will be referred to as a $\mathcal{P}$-example and a negative example as a $\mathcal{N}$-example. Given a training set $S = S_{\mathcal{P}} \cup S_{\mathcal{N}}$ of examples, the set of positive training examples will be denoted by $S_{\mathcal{P}}$ and the set of negative training examples by $S_{\mathcal{N}}$.

Any learning algorithm that constructs a conjunction (such as the one above) can be transformed into an algorithm constructing a disjunction just by exchanging the role of the positive and negative examples. Hence, simply by reassigning the set of negative training examples to the set of positive training examples (i.e., $S_{\mathcal{P}} \leftarrow S_{\mathcal{N}}$) and the set of positive training examples to the set of negative training examples (i.e., $S_{\mathcal{N}} \leftarrow S_{\mathcal{P}}$) we can transform the algorithm into one that constructs a disjunction. However, for the remainder of the paper we will assume, without loss of generality, that the SCM always produces a conjunction.

In this paper, we consider the case where $\mathcal{H}$ is the set of *data-dependent balls*.

Definition 1 *For each training example* $\mathbf{x}_i$ *with label* $y_i \in \{0,1\}$ *and (real-valued) radius* ρ*, we define feature* $h_{i,\rho}$ *to be the following* data-dependent ball *centered on* $\mathbf{x}_i$*:*

$$h_{i,\rho}(\mathbf{x}) \overset{\text{def}}{=} h_\rho(\mathbf{x},\mathbf{x}_i) = \begin{cases} y_i & \text{if } d(\mathbf{x},\mathbf{x}_i) < \rho \\ \bar{y}_i & \text{otherwise}\,, \end{cases}$$

where $\bar{y}_i$ *denotes the Boolean complement of* y_i *and* $d(\mathbf{x},\mathbf{x}')$ *denotes the distance between* $\mathbf{x}$ *and* $\mathbf{x}'$*. Training example* $\mathbf{x}_i$ *will be called the* ball center *of* $h_{i,\rho}$*.*

To determine the radius ρ *of ball* $h_{i,\rho}$*, we will use another training example* $\mathbf{x}_j$*, called the* ball border *of* $h_{i,\rho}$*, such that* $\rho = d(\mathbf{x}_j,\mathbf{x}_i)$*.*

Hence, the set $\mathcal{R} \subset \mathcal{H}$ of features used by the SCM gives us a set of ball centers and a set of ball borders. The union of these two sets gives the *compression set* of the SCM. Following Littlestone and Warmuth (1986) and Floyd and Warmuth (1995), the compression set is a small subset of the training set which identifies a classifier (here a SCM). The function that maps arbitrary compression sets to classifiers is called the *reconstruction function*. We refine further these notions in Section 4.

We adopt the PAC model where it is assumed that each example $(\mathbf{x},y)$ is drawn independently at random according to a fixed (but unknown) distribution. In this paper, we consider the probabilities of events taken separately over the $\mathcal{P}$-examples and the $\mathcal{N}$-examples. We will therefore denote by $P\{a(\mathbf{x},y)\big|(\mathbf{x},y)\in\mathcal{P}\}$ the probability that predicate a is true on a random draw of an example $(\mathbf{x},y)$, given that this example is positive. Hence, the error probability of classifier f on $\mathcal{P}$-examples and on $\mathcal{N}$-examples, that we call respectively the *expected $\mathcal{P}$-loss* and the *expected $\mathcal{N}$-loss*, are given by

$$\begin{aligned}\mathrm{er}_{\mathcal{P}}(f) &\stackrel{\mathrm{def}}{=} P\{f(\mathbf{x})\neq y\big|(\mathbf{x},y)\in\mathcal{P}\}\,,\\ \mathrm{er}_{\mathcal{N}}(f) &\stackrel{\mathrm{def}}{=} P\{f(\mathbf{x})\neq y\big|(\mathbf{x},y)\in\mathcal{N}\}\,.\end{aligned}$$

Similarly, let $\hat{\mathrm{er}}_{\mathcal{P}}(f,S)$ denote the number of examples in $S_{\mathcal{P}}$ misclassified by f and let $\hat{\mathrm{er}}_{\mathcal{N}}(f,S))$ denote the number of examples in $S_{\mathcal{N}}$ misclassified by f. Hence

$$\begin{aligned}\hat{\mathrm{er}}_{\mathcal{P}}(f,S) &\stackrel{\mathrm{def}}{=} |\{(\mathbf{x},y)\in S_{\mathcal{P}}:\ f(\mathbf{x})\neq y\}|\,,\\ \hat{\mathrm{er}}_{\mathcal{N}}(f,S) &\stackrel{\mathrm{def}}{=} \big|\{(\mathbf{x},y)\in S_{\mathcal{N}}:\ f(\mathbf{x})\neq y\}\big|\,.\end{aligned}$$

Throughout this paper, the probability of occurrence of a positive example will be denoted by $p_{\mathcal{P}}$. Similarly, $p_{\mathcal{N}}$ will denote the probability of occurrence of a negative example. We will consider the general case where the loss $l_{\mathcal{P}}$ of misclassifying a positive example can differ from the loss $l_{\mathcal{N}}$ of misclassifying a negative example. We will denote by $A(S)$ the classifier returned by the learning algorithm A trained on a set S of examples. In this case, the expected loss $\mathbb{E}[l(A(S))]$ of classifier $A(S)$ is defined as

$$\mathbb{E}[l(A(S))] \stackrel{\mathrm{def}}{=} l_{\mathcal{P}}\cdot p_{\mathcal{P}}\cdot \mathrm{er}_{\mathcal{P}}[A(S)] \,+\, l_{\mathcal{N}}\cdot p_{\mathcal{N}}\cdot \mathrm{er}_{\mathcal{N}}[A(S)]\,. \tag{1}$$

3. Incorrect Bound

The Theorem 5 of Marchand and Shawe-Taylor (2002) gives the following loss bound for the SCM with the symmetric loss case of $l_{\mathcal{P}} = l_{\mathcal{N}} = 1$.

Given the above definitions, let A be any learning algorithm that builds a SCM with data-dependent balls with the constraint that the returned function $A(S)$ always correctly classifies every example in the compression set. Then, with probability $1-\delta$ over all training sets S of m examples,

$$\mathbb{E}[l(A(S))] \;\leq\; 1-\exp\left\{-\frac{1}{m-c_p-b-c_n-k_p-k_n}\left(\ln B+\ln\frac{1}{\delta_0}\right)\right\},$$

where

$$\delta_0 \stackrel{\mathrm{def}}{=} \left(\frac{\pi^2}{6}\right)^{-5}\cdot((c_p+1)(c_n+1)(b+1)(k_p+1)(k_n+1))^{-2}\cdot\delta\,,$$

$$B \stackrel{\mathrm{def}}{=} \binom{m_p}{c_p}\binom{m_p-c_p}{b}\binom{m_n}{c_n}\binom{m_p-c_p-b}{k_p}\binom{m_n-c_n}{k_n}\,,$$

and where k_p and k_n are the number of misclassified positive and negative training examples by classifier $A(S)$. Similarly, c_p and c_n are the number of positive and negative ball centers contained in classifier $A(S)$ whereas b denotes the number of ball borders[1] *in classifier $A(S)$. Finally m_p and m_n denote the number of positive and negative examples in training set S.*

Let us take the B expression only and look more closely at the number of ways of choosing the errors on $S_{\mathcal{P}}$ and $S_{\mathcal{N}}$:

$$\binom{m_p - c_p - b}{k_p}\binom{m_n - c_n}{k_n}.$$

The bound on the expected loss given above will be small only if each factor is small. However, each factor can be small for a small number of training errors (desirable) or a large number of training errors (undesirable). In particular, the product of these two factors will be small for a small value of k_n (say, $k_n = 0$) and a large value of k_p (say, $k_p = m_p - c_p - b$). In this case, the denominator of the bound given above will become

$$m - c_p - b - c_n - k_p - k_n = m_n - c_n\,,$$

and will be large whenever $m_n \gg c_n$. Consequently, the bound given by Theorem 5 of Marchand and Shawe-Taylor (2002) will be small for classifiers having a small compression set and making a large number of errors on $S_{\mathcal{P}}$ and a small number of errors on $S_{\mathcal{N}}$. Clearly, this is incorrect as it implies a classifier with good generalization ability and so exposes an error in the proof. In order to derive a loss bound where the issue of imbalanced misclassifications can be handled, the errors for positive and negative examples must be bounded separately.

The error in the proof of Theorem 5 of Marchand and Shawe-Taylor (2002) occurs at the first equality used in their Equation 3. This equality is tantamount to writing that for any fixed classifier f:

$$\begin{aligned} & P\left\{S \in \mathcal{X}\colon\ \hat{\text{er}}(f,S) = 0 \;\middle|\; |S_{\mathcal{P}}| = m_p\right\} \\ & \qquad = (1-\text{er}_{\mathcal{P}}(f))^{m_p}(1-\text{er}_{\mathcal{N}}(f))^{m_n} \times \binom{m}{m_p} p_{\mathcal{P}}^{m_p}(1-p_{\mathcal{P}})^{m_n} \quad \text{(false)}\,, \end{aligned}$$

where $p_{\mathcal{P}}$ denotes the probability of occurrence of a $\mathcal{P}$-example. However this last equation is *false* since the probability on the left hand side is conditioned on the fact the $|S_{\mathcal{P}}| = m_p$. Hence, we have instead

$$P\left\{S \in \mathcal{X}\colon\ \hat{\text{er}}(f,S) = 0 \;\middle|\; |S_{\mathcal{P}}| = m_p\right\} = (1-\text{er}_{\mathcal{P}})^{m_p}(1-\text{er}_{\mathcal{N}})^{m_n}\,.$$

4. Sample-Compression Loss Bounds for Imbalanced Data

Recall that X denotes the input space. Let $\mathcal{X} = (X \times \{0,1\})^m$ be the set of training sets of size m with inputs from X. We consider any learning algorithm A having the property that, when trained on a training set $S \in \mathcal{X}$, A produces a classifier $A(S)$ which can be identified solely by a subset $\Lambda = \{\Lambda_{\mathcal{P}} \cup \Lambda_{\mathcal{N}}\} \subset S$, called the *compression set*, and a *message string* σ that represents some additional information required to obtain a classifier. Here $\Lambda_{\mathcal{P}}$ represents a subset of positive examples and $\Lambda_{\mathcal{N}}$

[1] As explained in Marchand and Shawe-Taylor (2002), the ball borders are always positive examples.

a subset of negative examples. More formally, this means that there exists a *reconstruction function* Φ that produces a classifier $f = \Phi(\Lambda, \sigma)$ when given an arbitrary compression set Λ and message string σ. We can thus consider that the learning algorithm A, trained on S, returns a compression set $\Lambda(S)$ and a message string $\sigma(S)$. The classifier is then given by $\Phi(\Lambda(S), \sigma(S))$.

For any training sample S and compression set Λ, consisting of a subset $\Lambda_{\mathcal{P}}$ of positive examples and a subset $\Lambda_{\mathcal{N}}$ of negative examples, we use the notation $\Lambda(S) = (\Lambda_{\mathcal{P}}(S), \Lambda_{\mathcal{N}}(S))$. Any further partitioning of the compression set Λ can be performed by the message string σ. For example, in the set covering machine, σ specifies for each point in $\Lambda_{\mathcal{P}}$, whether it is a ball center or a ball border (not already used as a center). As explained by Marchand and Shawe-Taylor (2002), this is the only additional information required to obtain a SCM consistent with the compression set.

We will use d_p to denote the number of examples present in $\Lambda_{\mathcal{P}}$. Similarly, d_n will denote the number of examples present in $\Lambda_{\mathcal{N}}$. To simplify the notation, we will use the $\mathbf{m}_{\mathcal{P}}$ and $\mathbf{m}_{\mathcal{N}}$ vectors defined as

$$\begin{aligned} \mathbf{m}_{\mathcal{P}} &\overset{\text{def}}{=} (m, m_p, m_n, d_p, d_n, k_p), \\ \mathbf{m}_{\mathcal{N}} &\overset{\text{def}}{=} (m, m_p, m_n, d_p, d_n, k_n), \end{aligned} \tag{2}$$

and

$$\mathbf{m}_{\mathcal{P}}(S, A(S)) \overset{\text{def}}{=} \big(|S|, |S_{\mathcal{P}}|, |S_{\mathcal{N}}|, |\Lambda_{\mathcal{P}}(S)|, |\Lambda_{\mathcal{N}}(S)|, \hat{\text{er}}_{\mathcal{P}}(A(S), S)\big), \tag{3}$$

$$\mathbf{m}_{\mathcal{N}}(S, A(S)) \overset{\text{def}}{=} \big(|S|, |S_{\mathcal{P}}|, |S_{\mathcal{N}}|, |\Lambda_{\mathcal{P}}(S)|, |\Lambda_{\mathcal{N}}(S)|, \hat{\text{er}}_{\mathcal{N}}(A(S), S)\big). \tag{4}$$

Hence, the predicate $\mathbf{m}_{\mathcal{P}}(S, A(S)) = \mathbf{m}_{\mathcal{P}}$ means that $|S| = m$, $|S_{\mathcal{P}}| = m_p$, $|S_{\mathcal{N}}| = m_n$, $|\Lambda_{\mathcal{P}}(S)| = d_p$, $|\Lambda_{\mathcal{N}}(S)| = d_n$, $\hat{\text{er}}_{\mathcal{P}}(A(S), S) = k_p$. We use a similar definition for predicate $\mathbf{m}_{\mathcal{N}}(S, A(S)) = \mathbf{m}_{\mathcal{N}}$. We will also use $B_{\mathcal{P}}(\mathbf{m}_{\mathcal{P}})$ and $B_{\mathcal{N}}(\mathbf{m}_{\mathcal{N}})$ defined as

$$\begin{aligned} B_{\mathcal{P}}(\mathbf{m}_{\mathcal{P}}) &\overset{\text{def}}{=} \binom{m_p}{d_p}\binom{m_n}{d_n}\binom{m_p - d_p}{k_p}, \\ B_{\mathcal{N}}(\mathbf{m}_{\mathcal{N}}) &\overset{\text{def}}{=} \binom{m_p}{d_p}\binom{m_n}{d_n}\binom{m_n - d_n}{k_n}. \end{aligned}$$

The proposed loss bound will hold uniformly for all possible messages that can be chosen by A. It will thus loosen as we increase the set $\mathcal{M}$ of possible messages that can be used. To obtain a smaller loss bound, we will therefore permit $\mathcal{M}$ to be *dependent* on the compression set chosen by A. In fact, the loss bound will depend on a *prior* distribution $P_\Lambda(\sigma)$ of message strings over the set $\mathcal{M}_\Lambda$ of possible messages that can be used with a compression set Λ. We will see that the only condition that P_Λ needs to satisfy is

$$\sum_{\sigma \in \mathcal{M}_\Lambda} P_\Lambda(\sigma) \leq 1.$$

Consider, for example, the case of a SCM conjunction of balls. Given a compression set $\Lambda = (\Lambda_{\mathcal{P}}, \Lambda_{\mathcal{N}})$ of size $(|\Lambda_{\mathcal{P}}|, |\Lambda_{\mathcal{N}}|) = (d_p, d_n)$, recall that each example in $\Lambda_{\mathcal{N}}$ is a ball center whereas each example in $\Lambda_{\mathcal{P}}$ can either be a ball border or a ball center. Hence, to specify a classifier given Λ, we only need to specify the examples in $\Lambda_{\mathcal{P}}$ that are ball borders.[2] This specification can be used

[2] For a SCM making no error with Λ, we can pair each center with its border in the following way. For each negative center, we choose the closest border. For each positive center, we choose the furthest border.

with a message string containing two parts. The first part specifies the number $b \in \{0,\ldots,d_p\}$ of ball borders in $\Lambda_{\mathcal{P}}$. The second part specifies which subset, among the set of $\binom{d_p}{b}$ possible subsets, is used for the set of ball borders. Consequently, if $b(\sigma)$ denotes the number of ball borders specified by message string σ, we can choose

$$P_\Lambda(\sigma) \;=\; \zeta(b(\sigma)) \cdot \binom{d_p}{b(\sigma)}^{-1} \quad \text{(SCM case)}\,, \tag{5}$$

where, for any non-negative integer b, we define

$$\zeta(b) \overset{\text{def}}{=} \frac{6}{\pi^2}(b+1)^{-2}\,. \tag{6}$$

Indeed, in this case, we clearly satisfy

$$\sum_{\sigma\in\mathcal{M}_\Lambda} P_\Lambda(\sigma) \;=\; \sum_{b=0}^{d_p} \zeta(b) \sum_{\sigma:b(\sigma)=b} \binom{d_p}{b(\sigma)}^{-1} \;\le\; 1\,.$$

The proposed loss bound will make use of the following functions:

$$\varepsilon_{\mathcal{P}}(\mathbf{m}_{\mathcal{P}},\beta) \quad \overset{\text{def}}{=} \quad 1-\exp\left(-\frac{1}{m_p-d_p-k_p}\left[\ln\left(B_{\mathcal{P}}(\mathbf{m}_{\mathcal{P}})\right)+\ln\frac{1}{\beta}\right]\right)\,, \tag{7}$$

$$\varepsilon_{\mathcal{N}}(\mathbf{m}_{\mathcal{N}},\beta) \quad \overset{\text{def}}{=} \quad 1-\exp\left(-\frac{1}{m_n-d_n-k_n}\left[\ln\left(B_{\mathcal{N}}(\mathbf{m}_{\mathcal{N}})\right)+\ln\frac{1}{\beta}\right]\right)\,. \tag{8}$$

Theorem 2 *Given the above definitions, let A be any learning algorithm having a reconstruction function that maps compression sets and message strings to classifiers. For any prior distribution* P_Λ *of messages and for any* $\delta \in (0,1]$*:*

$$P\Big\{S\in\mathcal{X}\,:\; \mathrm{er}_{\mathcal{P}}[A(S)] \le \varepsilon_{\mathcal{P}}\Big(\mathbf{m}_{\mathcal{P}}(S,A(S)), g_{\mathcal{P}}(S)\delta\Big)\Big\} \quad\ge\quad 1-\delta,$$

$$P\Big\{S\in\mathcal{X}\,:\; \mathrm{er}_{\mathcal{N}}[A(S)] \le \varepsilon_{\mathcal{N}}\Big(\mathbf{m}_{\mathcal{N}}(S,A(S)), g_{\mathcal{N}}(S)\delta\Big)\Big\} \quad\ge\quad 1-\delta,$$

where $\mathbf{m}_{\mathcal{P}}(S,A(S))$ *and* $\mathbf{m}_{\mathcal{N}}(S,A(S))$ *are defined by Equation 3 and Equation 4, and*

$$g_{\mathcal{P}}(S) \quad \overset{\text{def}}{=} \quad \zeta(d_p(S))\cdot\zeta(d_n(S))\cdot\zeta(k_p(S))\cdot P_{\Lambda(S)}(\sigma(S))\,, \tag{9}$$

$$g_{\mathcal{N}}(S) \quad \overset{\text{def}}{=} \quad \zeta(d_p(S))\cdot\zeta(d_n(S))\cdot\zeta(k_n(S))\cdot P_{\Lambda(S)}(\sigma(S))\,. \tag{10}$$

Note that Theorem 2 directly applies to the SCM when we use the distribution of messages given by Equation 5.

Proof To prove Theorem 2, it suffice to upper bound by δ the following probability

$$\begin{aligned} P \quad &\overset{\text{def}}{=} \quad P\Big\{S\in\mathcal{X}\,:\; \mathrm{er}_{\mathcal{P}}[A(S)] \ge \varepsilon\Big(\mathbf{m}_{\mathcal{P}}(S,A(S)),\Lambda(S),\sigma(S)\Big)\Big\} \\ &= \quad \sum_{\mathbf{m}_{\mathcal{P}}} P\Big\{S\in\mathcal{X}\,:\; \mathrm{er}_{\mathcal{P}}[A(S)] \ge \varepsilon\Big(\mathbf{m}_{\mathcal{P}},\Lambda(S),\sigma(S)\Big), \mathbf{m}_{\mathcal{P}}(S,A(S)) = \mathbf{m}_{\mathcal{P}}\Big\}\,, \end{aligned}$$

where $\varepsilon(\mathbf{m}_{\mathcal{P}}, \Lambda(S), \sigma(S))$ denotes a risk bound on $\mathrm{er}_{\mathcal{P}}(A(S))$ that depends (partly) on the compression set $\Lambda(S)$ and the message string $\sigma(S)$ returned by $A(S)$. The summation over $\mathbf{m}_{\mathcal{P}}$ stands for

$$\sum_{\mathbf{m}_{\mathcal{P}}}(\cdot) \overset{\text{def}}{=} \sum_{m_p=0}^{m} \sum_{d_p=0}^{m_p} \sum_{d_n=0}^{m-m_p} \sum_{k_p=0}^{m_p-d_p} (\cdot) \,.$$

Note that the summation over k_p stops at $m_p - d_p$ because, as we will see later in the proof, we can upper bound the risk of a sample-compressed classifier only from the training errors it makes on the examples that are not used for the compression set.

We will now use the notation $\mathbf{i} = (i_1, \ldots, i_d)$ for a sequence (or a vector) of strictly increasing indices, $0 < i_1 < i_2 < \cdots < i_d \leq m$. Hence there are 2^m distinct sequences $\mathbf{i}$. We will also use $|\mathbf{i}|$ to denote the length d of a sequence $\mathbf{i}$. Such sequences (or vectors) of indices will be used to identify subsets of S. For $S \in \mathcal{X}$, we define $S_{\mathbf{i}}$ as

$$S_{\mathbf{i}} \overset{\text{def}}{=} ((x_{i_1}, y_{i_1}), \ldots, (x_{i_d}, y_{i_d})) \;.$$

Under the constraint that $\mathbf{m}(S, A(S)) = \mathbf{m}$, we will denote by $\mathbf{i}_p$ any sequence (or vector) of indices where each index points to an example of $S_{\mathcal{P}}$. We also use an equivalent definition for $\mathbf{i}_n$. If, for example, $\mathbf{i}_n = (2,3,6,9)$, then $S_{\mathbf{i}_n}$ will denote the set of examples consisting of the second, third, sixth, and ninth $\mathcal{N}$-example of S. Therefore, given a training set S and vectors $\mathbf{i}_p$ and $\mathbf{i}_n$, the subset $S_{\mathbf{i}_p,\mathbf{i}_n}$ will denote a compression set. We will also denote by I_{m_p} the set of all the 2^{m_p} possible vectors $\mathbf{i}_p$ under the constraint that $|S_{\mathcal{P}}| = m_p$. We also use an equivalent definition for I_{m_n}. Using these definitions, we will now upper bound P uniformly over all possible realizations of $\mathbf{i}_p$ and $\mathbf{i}_n$ under the constraint $\mathbf{m}_{\mathcal{P}}(S, A(S)) = \mathbf{m}_{\mathcal{P}}$. Thus

$$\begin{aligned}
P \;\; &\leq \;\; \sum_{\mathbf{m}_{\mathcal{P}}} P\Big\{ S \in \mathcal{X} \colon \exists \mathbf{i}_p \in I_{m_p}, \exists \mathbf{i}_n \in I_{m_n}, \exists \sigma \in \mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}} : \\
&\qquad\qquad \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n}, \sigma)] \geq \varepsilon\Big(\mathbf{m}_{\mathcal{P}}, S_{\mathbf{i}_p,\mathbf{i}_n}, \sigma\Big), \mathbf{m}_{\mathcal{P}}(S, A(S)) = \mathbf{m}_{\mathcal{P}} \Big\} \\
&\leq \;\; \sum_{\mathbf{m}_{\mathcal{P}}} \sum_{\mathbf{i}_p \in I_{m_p}} \sum_{\mathbf{i}_n \in I_{m_n}} P\Big\{ S \in \mathcal{X} \;:\; \exists \sigma \in \mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}} : \\
&\qquad\qquad \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n}, \sigma)] \geq \varepsilon\Big(\mathbf{m}_{\mathcal{P}}, S_{\mathbf{i}_p,\mathbf{i}_n}, \sigma\Big), \mathbf{m}_{\mathcal{P}}(S, A(S)) = \mathbf{m}_{\mathcal{P}} \Big\},
\end{aligned}$$

where $\Phi(S_{\mathbf{i}_p,\mathbf{i}_n}, \sigma)$ denotes the classifier obtained once, $S, \mathbf{i}_p, \mathbf{i}_n$, and σ have been fixed. The last inequality comes from the union bound over all the possible choices of $\mathbf{i}_p \in I_{m_p}$ and $\mathbf{i}_n \in I_{m_n}$. Let

$$P' \overset{\text{def}}{=} P\Big\{ S \in \mathcal{X} \colon \exists \sigma \in \mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}} : \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n}, \sigma)] \geq \varepsilon\Big(\mathbf{m}_{\mathcal{P}}, S_{\mathbf{i}_p,\mathbf{i}_n}, \sigma\Big), \mathbf{m}_{\mathcal{P}}(S, A(S)) = \mathbf{m}_{\mathcal{P}} \Big\} \,.$$

We now make explicit how the positive and negative examples are interleaved in the training sequence S by introducing a new variable $\mathbf{b}$, which is a bit-string of length m such that S_i is a positive example if and only if $\mathbf{b}_i = 1$. Let B_{m_p} denote the set of possible $\mathbf{b}$ vectors that we can have under the constraint that $|S_{\mathcal{P}}| = m_p$. We then have

$$\begin{aligned} P' &= \sum_{\mathbf{b}\in B_{m_p}} P\Big\{S\in\mathcal{X}:\ \exists\sigma\in\mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}}: \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)] \ge \varepsilon\Big(\mathbf{m}_{\mathcal{P}}, S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big), \\ &\qquad \mathbf{m}_{\mathcal{P}}(S,A(S)) = \mathbf{m}_{\mathcal{P}} \mid\ \mathbf{b}(S)=\mathbf{b}\Big\} P\Big\{S\in\mathcal{X}:\ \mathbf{b}(S)=\mathbf{b}\Big\} \\ &= \sum_{\mathbf{b}\in B_{m_p}} P\Big\{S\in\mathcal{X}:\ \exists\sigma\in\mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}}: \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)] \ge \varepsilon\Big(\mathbf{m}_{\mathcal{P}}, S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big), \\ &\qquad \mathbf{m}_{\mathcal{P}}(S,A(S)) = \mathbf{m}_{\mathcal{P}} \mid\ \mathbf{b}(S)=\mathbf{b}\Big\} p_{\mathcal{P}}^{m_p}(1-p_{\mathcal{P}})^{m-m_p}. \end{aligned}$$

$$\begin{aligned} P' &\le \binom{m}{m_p} p_{\mathcal{P}}^{m_p}(1-p_{\mathcal{P}})^{m-m_p} \sup_{\mathbf{b}\in B_{m_p}} P\Big\{S\in\mathcal{X}:\ \exists\sigma\in\mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}}: \\ &\qquad \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)] \ge \varepsilon\Big(\mathbf{m}_{\mathcal{P}}, S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big), \mathbf{m}_{\mathcal{P}}(S,A(S)) = \mathbf{m}_{\mathcal{P}} \mid\ \mathbf{b}(S)=\mathbf{b}\Big\}. \end{aligned}$$

Under the condition $\mathbf{b}(S)=\mathbf{b}$, index vectors $\mathbf{i}_p$ and $\mathbf{i}_n$ are now pointing to specific examples in S. Consequently, under this condition, we can compute the above probability by first conditioning on the compression set $S_{\mathbf{i}_p,\mathbf{i}_n}$ and then performing the expectation over $S_{\mathbf{i}_p,\mathbf{i}_n}$. Hence

$$P\Big\{S\in\mathcal{X}\colon(\cdot)\Big|\mathbf{b}(S)=\mathbf{b}\Big\} \ = \ E_{S_{\mathbf{i}_p,\mathbf{i}_n}|\mathbf{b}} P\Big\{S\in\mathcal{X}\colon(\cdot)\Big|\mathbf{b}(S)=\mathbf{b}, S_{\mathbf{i}_p,\mathbf{i}_n}\Big\}.$$

By applying the union bound over $\sigma\in\mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}}$, we obtain

$$\begin{aligned} &P\Big\{S\in\mathcal{X}\colon \exists\sigma\in\mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}}: \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)] \ge \varepsilon\Big(\mathbf{m}_{\mathcal{P}}, S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big), \\ &\qquad \mathbf{m}_{\mathcal{P}}(S,A(S)) = \mathbf{m}_{\mathcal{P}} \mid\ \mathbf{b}(S)=\mathbf{b}, S_{\mathbf{i}_p,\mathbf{i}_n}\Big\} \\ &\le \sum_{\sigma\in\mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}}} P\Big\{S\in\mathcal{X}\colon \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)] \ge \varepsilon\Big(\mathbf{m}_{\mathcal{P}}, S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big), \\ &\qquad \mathbf{m}_{\mathcal{P}}(S,A(S)) = \mathbf{m}_{\mathcal{P}} \mid\ \mathbf{b}(S)=\mathbf{b}, S_{\mathbf{i}_p,\mathbf{i}_n}\Big\}. \end{aligned}$$

We will now stratify this last probability by the set of possible errors that classifier $\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)$ can perform on the training examples that are not in the compression set $S_{\mathbf{i}_p,\mathbf{i}_n}$. Note that we do not force here the learner to produce a classifier that does not make errors on $S_{\mathbf{i}_p,\mathbf{i}_n}$. However, the set of message strings needed by Φ to identify a classifier h might be larger when h can err on $S_{\mathbf{i}_p,\mathbf{i}_n}$. To perform this stratification, let $\hat{\mathbf{er}}(f,S_{\mathcal{P}})$ be the vector of indices pointing to the examples of $S_{\mathcal{P}}$ that are misclassified by f. Moreover, let $I_{m_p}(\mathbf{i}_p)$ denote the set of all vectors $\mathbf{j}_p\in I_{m_p}$ for which no index $i\in\mathbf{j}_p$ is also in $\mathbf{i}_p$. In other words, for all $\mathbf{i}_p\in I_{m_p}$ and all $\mathbf{j}_p\in I_{m_p}(\mathbf{i}_p)$, we have $\mathbf{j}_p\cap\mathbf{i}_p=\emptyset$.

Therefore

$$\begin{aligned}
&P\Big\{S\in\mathcal{X}\colon \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)]\geq\varepsilon\Big(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big),\mathbf{m}_{\mathcal{P}}(S,A(S))=\mathbf{m}_{\mathcal{P}}\ \mid\ \mathbf{b}(S)=\mathbf{b},S_{\mathbf{i}_p,\mathbf{i}_n}\Big\}\\
&\qquad=\sum_{\mathbf{j}_p\in I_{m_p}(\mathbf{i}_p)}P\Big\{S\in\mathcal{X}\colon \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)]\geq\varepsilon\Big(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big),\\
&\qquad\qquad\hat{\mathbf{er}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma),S_{\mathcal{P}}]=\mathbf{j}_p,\mathbf{m}_{\mathcal{P}}(S,A(S))=\mathbf{m}_{\mathcal{P}}\ \mid\ \mathbf{b}(S)=\mathbf{b},S_{\mathbf{i}_p,\mathbf{i}_n}\Big\}\\
&\qquad=\sum_{\mathbf{j}_p\in I_{m_p}(\mathbf{i}_p)}P\Big\{S\in\mathcal{X}\colon \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)]\geq\varepsilon\Big(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big),\\
&\qquad\qquad\qquad\qquad\hat{\mathbf{er}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma),S_{\mathcal{P}}]=\mathbf{j}_p\ \mid\ \mathbf{b}(S)=\mathbf{b},S_{\mathbf{i}_p,\mathbf{i}_n}\Big\},
\end{aligned}$$

where the last equality comes from the fact that the condition $\mathbf{m}_{\mathcal{P}}(S,A(S))=\mathbf{m}_{\mathcal{P}}$ is obsolete when $\mathbf{b}(S)=\mathbf{b}$ with fixed vectors $\mathbf{i}_p,\mathbf{i}_n,\mathbf{j}_p$. Now, under the condition $\mathbf{b}(S)=\mathbf{b}$ with a fixed compression set $S_{\mathbf{i}_p,\mathbf{i}_n}$, this last probability is obtained for the random draws of the training examples that are not in $S_{\mathbf{i}_p,\mathbf{i}_n}$. Consequently, this last probability is at most equal to the probability that a fixed classifier, having $\mathrm{er}_{\mathcal{P}}\geq\varepsilon(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)$, makes no errors on $m_p-d_p-k_p$ positive examples that are not in the compression set $S_{\mathbf{i}_p,\mathbf{i}_n}$. Note that the probability space created by the conditioning specifies only the positions of the positive examples but places no further restrictions on them. They can therefore be viewed as independent draws from the distribution of positive examples. This makes it possible to bound the probability of the event by the probability that $m_p-d_p-k_p$ independent draws are all correctly classified. Hence, we have

$$\begin{aligned}
&P\Big\{S\in\mathcal{X}\colon \mathrm{er}_{\mathcal{P}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)]\geq\varepsilon\Big(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma\Big),\\
&\qquad\qquad\hat{\mathbf{er}}[\Phi(S_{\mathbf{i}_p,\mathbf{i}_n},\sigma),S_{\mathcal{P}}]=\mathbf{j}_p\ \mid\ \mathbf{b}(S)=\mathbf{b},S_{\mathbf{i}_p,\mathbf{i}_n}\Big\}\\
&\qquad\qquad\qquad\leq\ \Big(1-\varepsilon(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)\Big)^{m_p-d_p-k_p}.
\end{aligned}$$

By regrouping the previous results, we get

$$\begin{aligned}
P\ &\leq\ \sum_{\mathbf{m}_{\mathcal{P}}}\binom{m}{m_p}p_{\mathcal{P}}^{m_p}(1-p_{\mathcal{P}})^{m-m_p}\sum_{\mathbf{i}_p\in I_{m_p}}\sum_{\mathbf{i}_n\in I_{m_n}}\\
&\qquad\sup_{\mathbf{b}\in B_{m_p}}E_{S_{\mathbf{i}_p,\mathbf{i}_n}|\mathbf{b}}\sum_{\sigma\in\mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}}}\ \sum_{\mathbf{j}_p\in I_{m_p}(\mathbf{i}_p)}\Big(1-\varepsilon(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)\Big)^{m_p-d_p-k_p}\\
&=\ \sum_{m_p=0}^{m}\binom{m}{m_p}p_{\mathcal{P}}^{m_p}(1-p_{\mathcal{P}})^{m-m_p}\sum_{d_p=0}^{m_p}\binom{m_p}{d_p}\sum_{d_n=0}^{m-m_p}\binom{m_n}{d_n}\sum_{k_p=0}^{m_p-d_p}\binom{m_p-d_p}{k_p}\\
&\qquad\sup_{\mathbf{b}\in B_{m_p}}E_{S_{\mathbf{i}_p,\mathbf{i}_n}|\mathbf{b}}\sum_{\sigma\in\mathcal{M}_{S_{\mathbf{i}_p,\mathbf{i}_n}}}\Big(1-\varepsilon(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)\Big)^{m_p-d_p-k_p}.
\end{aligned}$$

By using

$$\left(1-\varepsilon(\mathbf{m}_{\mathcal{P}},S_{\mathbf{i}_p,\mathbf{i}_n},\sigma)\right)^{m_p-d_p-k_p} \;=\; P_{S_{\mathbf{i}_p,\mathbf{i}_n}}(\sigma)\cdot\frac{1}{B_{\mathcal{P}}(\mathbf{m}_{\mathcal{P}})}\cdot\zeta(k_p)\cdot\zeta(d_n)\cdot\zeta(d_p)\cdot\delta\,,$$

we get $P\leq\delta$ as desired. Similarly, we have

$$P\left\{S\in\mathcal{X}\;:\;\mathrm{er}_{\mathcal{N}}[A(S)]\geq\varepsilon_{\mathcal{N}}\Big(\mathbf{m}_{\mathcal{N}}(S,A(S)),g_{\mathcal{N}}(S)\delta\Big)\right\}\;\leq\;\delta,$$

which completes the proof. ■

Remark 3 *This theorem can be viewed in a standard asymptotic form by using the inequality* $1-\exp(-x)\leq x$*, for* $x\geq 0$*. To see this, we simply need to substitute Equations 7 and 8 into each probability given in Theorem 2 and weaken them with the above inequality. Doing so yields the following bounds:*

$$P\left\{S\in\mathcal{X}\;:\;\mathrm{er}_{\mathcal{P}}[A(S)]\leq\frac{1}{m_p-d_p-k_p}\left[\ln\left(B_{\mathcal{P}}(\mathbf{m}_{\mathcal{P}})\right)+\ln\frac{1}{g_{\mathcal{P}}(S)\delta}\right]\right\}\quad\geq\quad 1-\delta,$$

$$P\left\{S\in\mathcal{X}\;:\;\mathrm{er}_{\mathcal{N}}[A(S)]\leq\frac{1}{m_n-d_n-k_n}\left[\ln\left(B_{\mathcal{N}}(\mathbf{m}_{\mathcal{N}})\right)+\ln\frac{1}{g_{\mathcal{N}}(S)\delta}\right]\right\}\quad\geq\quad 1-\delta.$$

However, each probability is separately bounding the error on the positive and negative examples and so will not (in the final bound) hold with probability $1-\delta$ *but with probability* $1-\delta/4$ *(to be shown) as the expected loss will rely on four bounds simultaneously holding true (i.e., from Equation 1 we would like to upper bound* $\mathrm{er}_{\mathcal{P}}[A(S)]$, $\mathrm{er}_{\mathcal{N}}[A(S)]$, $p_{\mathcal{P}}$ *and* $p_{\mathcal{N}}$*).*

Now that we have a bound on both $\mathrm{er}_{\mathcal{P}}[A(S)]$ and $\mathrm{er}_{\mathcal{N}}[A(S)]$, to bound the expected loss $\mathbb{E}[l(A(S))]$ of Equation 1 we now need to upper bound the probabilities $p_{\mathcal{P}}$ and $p_{\mathcal{N}}$. For this task, we could use a well-known approximation of the binomial tail such as the additive Hoeffding bound or the multiplicative Chernoff bound. However, the Hoeffding bound is known to be very loose when the the probability of interest (here $p_{\mathcal{P}}$ and $p_{\mathcal{N}}$) is close to zero. Conversely, the multiplicative Chernoff bound is known to be loose when the probability of interest is close to $1/2$. In order to obtain a tight loss bound for both balanced and imbalanced data sets, we have decided to use the binomial distribution without any approximation.

Recall that the probability $\mathrm{Bin}(m,k,p)$ of having at most k successes among m Bernoulli trials, each having probability of success p, is given by the binomial tail

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

Following Langford (2005), we now define the *binomial tail inversion* $\overline{\mathrm{Bin}}(m,k,\delta)$ as the largest value of probability of success such that we still have a probability of at least δ of observing at most k successes out of m Bernoulli trials. In other words,

$$\overline{\mathrm{Bin}}(m,k,\delta)\stackrel{\mathrm{def}}{=}\sup\Big\{p:\mathrm{Bin}(m,k,p)\geq\delta\Big\}\,. \tag{11}$$

From this definition, it follows that $\overline{\mathrm{Bin}}(m,m_n,\delta)$ is the *smallest* upper bound on $p_{\mathcal{N}}$, which holds with probability at least $1-\delta$, over the random draws of m examples. Hence

$$P\left\{S\in\mathcal{X}:\ p_{\mathcal{N}}\le\overline{\mathrm{Bin}}\Big(m,m_n,\delta\Big)\right\}\ge 1-\delta\,.$$

From this bound (applied to both $p_{\mathcal{P}}$ and $p_{\mathcal{N}}$), and from the previous theorem, the following predicates hold simultaneously with probability $1-\delta$ over the random draws of S:

$$\begin{aligned}
\mathrm{er}_{\mathcal{P}}[A(S)] &\le \varepsilon_{\mathcal{P}}\Big(\mathbf{m}_{\mathcal{P}},g_{\mathcal{P}}(S)\frac{\delta}{4}\Big)\,,\\
\mathrm{er}_{\mathcal{N}}[A(S)] &\le \varepsilon_{\mathcal{N}}\Big(\mathbf{m}_{\mathcal{N}},g_{\mathcal{N}}(S)\frac{\delta}{4}\Big)\,,\\
p_{\mathcal{N}} &\le \overline{\mathrm{Bin}}\Big(m,m_n,\frac{\delta}{4}\Big)\,,\\
p_{\mathcal{P}} &\le \overline{\mathrm{Bin}}\Big(m,m_p,\frac{\delta}{4}\Big)\,,
\end{aligned}$$

where $\mathbf{m}_{\mathcal{P}}=\mathbf{m}_{\mathcal{P}}(S,A(S))$ and $\mathbf{m}_{\mathcal{N}}=\mathbf{m}_{\mathcal{N}}(S,A(S))$. Consequently, we have the next theorem.

Theorem 4 *Given the above definitions, let A be any learning algorithm having a reconstruction function that maps compression sets and message strings to classifiers. With probability* $1-\delta$ *over the random draws of a training set S, we have*

$$\begin{aligned}
\mathbb{E}[l(A(S))]\ \le\ & l_{\mathcal{P}}\cdot\overline{\mathrm{Bin}}\Big(m,m_p,\frac{\delta}{4}\Big)\cdot\varepsilon_{\mathcal{P}}\Big(\mathbf{m}_{\mathcal{P}},g_{\mathcal{P}}(S)\frac{\delta}{4}\Big)\\
&+l_{\mathcal{N}}\cdot\overline{\mathrm{Bin}}\Big(m,m_n,\frac{\delta}{4}\Big)\cdot\varepsilon_{\mathcal{N}}\Big(\mathbf{m}_{\mathcal{N}},g_{\mathcal{N}}(S)\frac{\delta}{4}\Big)\,,
\end{aligned}$$

where $\mathbf{m}_{\mathcal{P}}=\mathbf{m}_{\mathcal{P}}(S,A(S))$ *and* $\mathbf{m}_{\mathcal{N}}=\mathbf{m}_{\mathcal{N}}(S,A(S))$ *are defined by Equations 3 and 4.*

We can now improve the loss bound given by Theorem 4 in the following way. Consider the frequencies $\hat{p}_{\mathcal{P}}\overset{\mathrm{def}}{=}m_p/m$ and $\hat{p}_{\mathcal{N}}\overset{\mathrm{def}}{=}m_n/m$. Let us simply denote by $\varepsilon_{\mathcal{P}}$ and $\varepsilon_{\mathcal{N}}$ some upper bounds on $\mathrm{er}_{\mathcal{P}}[A(S)]$ and $\mathrm{er}_{\mathcal{P}}[A(S)]$. Let us also denote by $\overline{p}_{\mathcal{P}}$ and $\overline{p}_{\mathcal{N}}$ some upper bounds on $p_{\mathcal{P}}$ and $p_{\mathcal{N}}$. Let us first assume that $l_{\mathcal{N}}\varepsilon_{\mathcal{N}}\ge l_{\mathcal{P}}\varepsilon_{\mathcal{P}}$. Then we have

$$\begin{aligned}
\mathbb{E}[l(A(S))] &\le p_{\mathcal{P}}l_{\mathcal{P}}\varepsilon_{\mathcal{P}}+p_{\mathcal{N}}l_{\mathcal{N}}\varepsilon_{\mathcal{N}}\\
&= l_{\mathcal{P}}\varepsilon_{\mathcal{P}}+p_{\mathcal{N}}(l_{\mathcal{N}}\varepsilon_{\mathcal{N}}-l_{\mathcal{P}}\varepsilon_{\mathcal{P}})\\
&\le l_{\mathcal{P}}\varepsilon_{\mathcal{P}}+\overline{p}_{\mathcal{N}}(l_{\mathcal{N}}\varepsilon_{\mathcal{N}}-l_{\mathcal{P}}\varepsilon_{\mathcal{P}})\\
&= \hat{p}_{\mathcal{P}}l_{\mathcal{P}}\varepsilon_{\mathcal{P}}+\hat{p}_{\mathcal{N}}l_{\mathcal{N}}\varepsilon_{\mathcal{N}}+(\overline{p}_{\mathcal{N}}-\hat{p}_{\mathcal{N}})(l_{\mathcal{N}}\varepsilon_{\mathcal{N}}-l_{\mathcal{P}}\varepsilon_{\mathcal{P}})\,.
\end{aligned}$$

Likewise, if $l_{\mathcal{P}}\varepsilon_{\mathcal{P}}\ge l_{\mathcal{N}}\varepsilon_{\mathcal{N}}$, we have

$$\mathbb{E}[l(A(S))]\le\hat{p}_{\mathcal{P}}l_{\mathcal{P}}\varepsilon_{\mathcal{P}}+\hat{p}_{\mathcal{N}}l_{\mathcal{N}}\varepsilon_{\mathcal{N}}+(\overline{p}_{\mathcal{P}}-\hat{p}_{\mathcal{P}})(l_{\mathcal{P}}\varepsilon_{\mathcal{P}}-l_{\mathcal{N}}\varepsilon_{\mathcal{N}})\,.$$

Consequently, we have the following theorem.

Theorem 5 *Given the above definitions, let A be any learning algorithm having a reconstruction function that maps compression sets and message strings to classifiers. For any real numbers a,b,c, let*

$$\Psi(a;b;c) \stackrel{\text{def}}{=} \begin{cases} a\cdot|c| & \text{if} \quad c \geq 0 \\ b\cdot|c| & \text{if} \quad c \leq 0\,. \end{cases}$$

Then, with probability $1-\delta$ over the random draws of a training set S, we have

$$\begin{aligned}\mathbb{E}[l(A(S))] \ \leq\ & \frac{m_p}{m}\cdot l_{\mathcal{P}}\cdot\varepsilon_{\mathcal{P}}\Big(\mathbf{m}_{\mathcal{P}}, g_{\mathcal{P}}(S)\frac{\delta}{4}\Big) + \frac{m_n}{m}\cdot l_{\mathcal{N}}\cdot\varepsilon_{\mathcal{N}}\Big(\mathbf{m}_{\mathcal{N}}, g_{\mathcal{N}}(S)\frac{\delta}{4}\Big) \\ & + \Psi\Bigg(\overline{\text{Bin}}\Big(m, m_p, \frac{\delta}{4}\Big) - \frac{m_p}{m}\ ;\ \overline{\text{Bin}}\Big(m, m_n, \frac{\delta}{4}\Big) - \frac{m_n}{m}\ ; \\ & \qquad l_{\mathcal{P}}\varepsilon_{\mathcal{P}}\Big(\mathbf{m}_{\mathcal{P}}, g_{\mathcal{P}}(S)\frac{\delta}{4}\Big) - l_{\mathcal{N}}\varepsilon_{\mathcal{N}}\Big(\mathbf{m}_{\mathcal{N}}, g_{\mathcal{N}}(S)\frac{\delta}{4}\Big)\Bigg)\,,\end{aligned}$$

where $\mathbf{m}_{\mathcal{P}} = \mathbf{m}_{\mathcal{P}}(S, A(S))$ and $\mathbf{m}_{\mathcal{N}} = \mathbf{m}_{\mathcal{N}}(S, A(S))$ are defined by Equations 3 and 4.

To compare the bound given by Theorem 5 with the bound given by Theorem 4, let us assume that $l_{\mathcal{N}}\varepsilon_{\mathcal{N}} \geq l_{\mathcal{P}}\varepsilon_{\mathcal{P}}$. Using our shorthand notation, the bound of Theorem 5 is given by

$$l_{\mathcal{P}}\hat{p}_{\mathcal{P}}\varepsilon_{\mathcal{P}} + l_{\mathcal{N}}\hat{p}_{\mathcal{N}}\varepsilon_{\mathcal{N}} + (\overline{p}_{\mathcal{N}} - \hat{p}_{\mathcal{N}})(l_{\mathcal{N}}\varepsilon_{\mathcal{N}} - l_{\mathcal{P}}\varepsilon_{\mathcal{P}})\,.$$

Whereas the bound of Theorem 4 is given by

$$l_{\mathcal{P}}\overline{p}_{\mathcal{P}}\varepsilon_{\mathcal{P}} + l_{\mathcal{N}}\overline{p}_{\mathcal{N}}\varepsilon_{\mathcal{N}}\,.$$

The bound of Theorem 4 minus the bound of Theorem 5 then gives

$$\begin{aligned}(l_{\mathcal{P}}\overline{p}_{\mathcal{P}}\varepsilon_{\mathcal{P}} + l_{\mathcal{N}}\overline{p}_{\mathcal{N}}\varepsilon_{\mathcal{N}}) - (l_{\mathcal{P}}\hat{p}_{\mathcal{P}}\varepsilon_{\mathcal{P}} + l_{\mathcal{N}}\hat{p}_{\mathcal{N}}\varepsilon_{\mathcal{N}} + (\overline{p}_{\mathcal{N}} - \hat{p}_{\mathcal{N}})(l_{\mathcal{N}}\varepsilon_{\mathcal{N}} - l_{\mathcal{P}}\varepsilon_{\mathcal{P}})) \\ = \ (\overline{p}_{\mathcal{P}} - \hat{p}_{\mathcal{P}})l_{\mathcal{P}}\varepsilon_{\mathcal{P}} + (\overline{p}_{\mathcal{N}} - \hat{p}_{\mathcal{N}})l_{\mathcal{N}}\varepsilon_{\mathcal{N}} - (\overline{p}_{\mathcal{N}} - \hat{p}_{\mathcal{N}})(l_{\mathcal{N}}\varepsilon_{\mathcal{N}} - l_{\mathcal{P}}\varepsilon_{\mathcal{P}}) \\ = \ (\overline{p}_{\mathcal{P}} - \hat{p}_{\mathcal{P}} + \overline{p}_{\mathcal{N}} - \hat{p}_{\mathcal{N}})l_{\mathcal{P}}\varepsilon_{\mathcal{P}} \\ = \ (\overline{p}_{\mathcal{P}} + \overline{p}_{\mathcal{N}} - 1)l_{\mathcal{P}}\varepsilon_{\mathcal{P}}\,.\end{aligned}$$

Since $l_{\mathcal{P}}\varepsilon_{\mathcal{P}} > 0$ and $\overline{p}_{\mathcal{P}} + \overline{p}_{\mathcal{N}} > 1$, we have an improvement using Theorem 5.

Example 1 *If $l_{\mathcal{P}} = l_{\mathcal{N}} = 1, \delta = 0.05, m = 100, m_p = 40, m_n = 60, \varepsilon_{\mathcal{P}} = 0.3, \varepsilon_{\mathcal{N}} = 0.4$, we get 0.439 for the bound of Theorem 4 and only 0.371 for the bound of Theorem 5. Hence, the bound of Theorem 5 can be significantly better than the the bound of Theorem 4.*

5. Discussion and Numerical Comparisons with Other Bounds

Let us first discuss the bounds that we have proposed and make explicit some of the details and consequences. In general, risk bounds are simply upper bounds of the true error calculated from the (overall) error achieved during training. There is no distinction made between the positive and negative class. The results of the current paper are bounds on the error achieved separately on the positive and negative examples. Hence, making the distinction between the two classes explicit. Furthermore, the risk bound on one class depends on what the classifier achieves on the training examples of that class. Thus, making the bound more data-dependent then the usual bounds on the true error. This strong data-dependence also allows the user to take into account the observed number of positive and negative examples in the training sample as well as the flexibility of specifying different losses for each class. This is known as asymmetric loss and is not possible with the current crop of sample-compression loss bounds.

Note also that the proposed bounds are data dependent bounds for which there are no corresponding lower bounds. A small compression scheme is evidence of simplicity in the structure of the classifier, but one that is related to the training distribution rather than a priori determined.

Any algorithm that uses a compression scheme can use the bounds that we have proposed and take advantage of asymmetrical loss and cases of imbalanced data sets. However, the tightness of the bound relies on the sparsity of the classifiers (e.g., the size of the compression set). Hence, it may not be advantageous to use algorithms that do not possess levels of sparsity similar (or comparable) to the SCM. This is one reason why we will provide a numerical comparison of various sample-compression bounds for the case of the SCM.

In order to show the merits of our bound we must now compare numerically against more common sample compression bounds and the bound found to be incorrect. In doing so we point out when our bound can be smaller and when it can become larger. All the compared bounds are specialized to the set covering machine compression scheme that uses data-dependent balls. Here each ball is constructed from two data points—one that defines the center of the ball and another that helps define the radius of the ball (known as the border point). Hence to build a classifier from the compression set, we also need an informative message string to discriminate between the border points and the centers.

Let us now discuss the experimental setup, including a list of all the bounds compared, and then conclude with a review of the results.

5.1 Setup

From Example 1 of Section 4, it is clear that using Theorem 5 is more advantageous than Theorem 4. Hence, all experiments will be conducted with the bound of Theorem 5. The first bound we compare against is taken from the original set covering machine paper by Marchand and Shawe-Taylor (2001) and is similar to the Littlestone and Warmuth (1986) bound but with more specialization for the SCM compression set defined from the set of data-dependent balls. The second generalization error bound is adapted from Marchand and Sokolova (2005) and is a slight modification of the Marchand and Shawe-Taylor (2001) result. All these bounds will also be compared against the incorrect bound given in Marchand and Shawe-Taylor (2002).

Please note that traditional sample compression bounds, such as that given by Theorem 6.1 of Langford (2005), cannot be used with the set covering machine as it does *not* allow the inclusion of any side information in the reconstruction of the classifier. The SCM, however, stores both the center and border points in order to construct its hypotheses. This implies the need for side information to discriminate between centers and border points, something that traditional sample compression bounds do not cater for. Therefore, we cannot give numerical comparisons against these types of bounds.

All generalization error bounds detailed below will make use of the following definitions: $d_n = c_n$, $d_p = c_p + b$, $d = d_p + d_n$ and $k = k_p + k_n$. For completeness, we give the definitions of all risk bounds not already stated and, to avoid repetition, we only give references to the bounds described earlier.

- **new bound** (Theorem 5). When applied to the SCM, the new bound uses the distribution of messages given by Equation 5 and Equations 6, 7, 8, 9, 10, and 11.
- **incorrect bound** (Theorem 5 of Marchand and Shawe-Taylor, 2002). This bound can also be found in Section 3 of the current paper.
- **MS01 bound** (Theorem 5.2 of Marchand and Shawe-Taylor, 2001):
$$\begin{aligned}\varepsilon(m,d,c_p,k,\delta) &= 1-\exp\left(\frac{-1}{m-2d-k}\left[\ln\binom{m}{2d}+\ln\binom{2d}{c_p}+\ln\binom{m-2d}{k}+\right.\right.\\ &\qquad\left.\left.\ln\left(\frac{2m^2d}{\delta}\right)\right]\right).\end{aligned}$$
- **MS05 bound** (Equation 10 of Marchand and Sokolova, 2005):
$$\begin{aligned}\varepsilon(m,d,d_p,b,k,\delta) &= 1-\exp\left(\frac{-1}{m-d-k}\left[\ln\binom{m}{d}+\ln\binom{m-d}{k}+\ln\binom{d_p}{b}+\right.\right.\\ &\qquad\left.\left.\ln\left(\frac{1}{\zeta(d)\zeta(k)\zeta(b)\delta}\right)\right]\right),\end{aligned}$$
where $\zeta(a)$ is given by Equation 6.

5.2 Discussion of Results

The numerical comparisons of these four bounds (*new bound, incorrect bound, MS01 bound and MS05 bound*) are shown in Figure 1 and Figure 2. Each plot contains the number of positive examples m_p, the number of negative examples m_n, the number of positive centers c_p, the number of negative centers c_n and the number of borders b. The number of negative misclassifications k_n was fixed for all plots and these values can be found in the x-axis label (either 0 or 500). The number of positive examples was varied and its quantity was set to those values given by the x-axis of the plot. For example, in the left hand side plot of Figure 1, the number of negative misclassifications k_n was 0 and the number of positive misclassifications k_p varied from 1 to 2000. The y-axis give the bound values achieved. Finally, the empirical error was also included in each plot—which is simply the number of examples misclassified divided by the number of examples, that is, $(k_p+k_n)/(m_p+m_n)$.

Figure 1 shows the case where the number of positive and negative examples is approximately the same. We clearly see that the incorrect bound becomes erroneous when the number k_p of errors on the positive training examples approaches the total number m_p of positive training examples. We also see that the new bound is tighter than the MS01 and MS05 bounds when the k_p differs greatly from k_n. However, the latter bound is slightly tighter than the new bound when $k_p = k_n$.

Figure 2 depicts the case where there is an imbalance in the data set ($m_n \gg m_p$), implying greater possibility of imbalance in misclassifications. However, the behavior is similar as the one found in Figure 1. Indeed, the MS01 and MS05 loss bounds are slightly smaller than the new bound when k_p/m_p is similar to k_n/m_n, but the new bound becomes smaller when these two quantities greatly differ. This is where the new bound is most advantageous—in the case when there is an imbalance in misclassifications. As we would expect, the new bound is smaller when one class of examples is more abundant than the other.

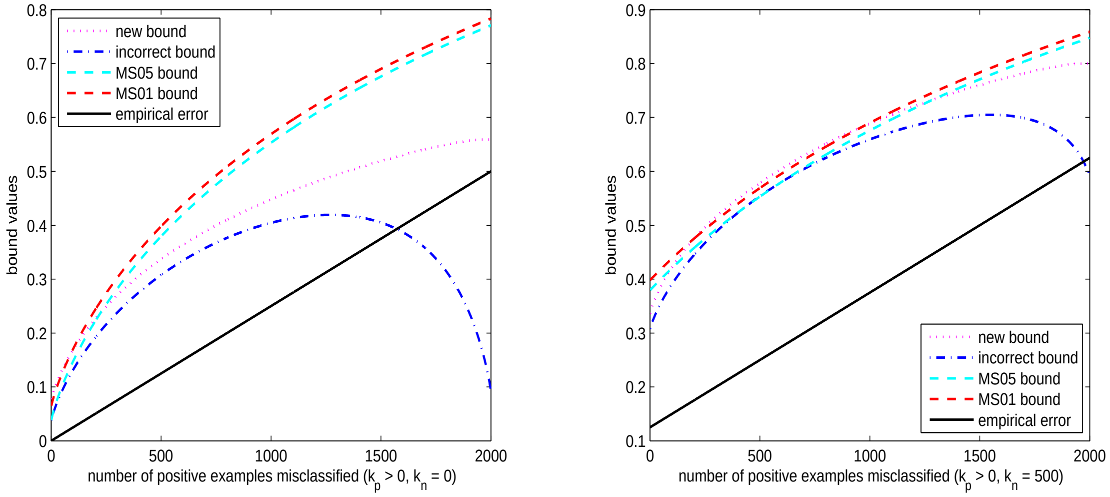

Figure 1: Bound values for the SCM when $m_p = 2020, m_n = 1980, c_p = 5, c_n = 5, b = 10$.

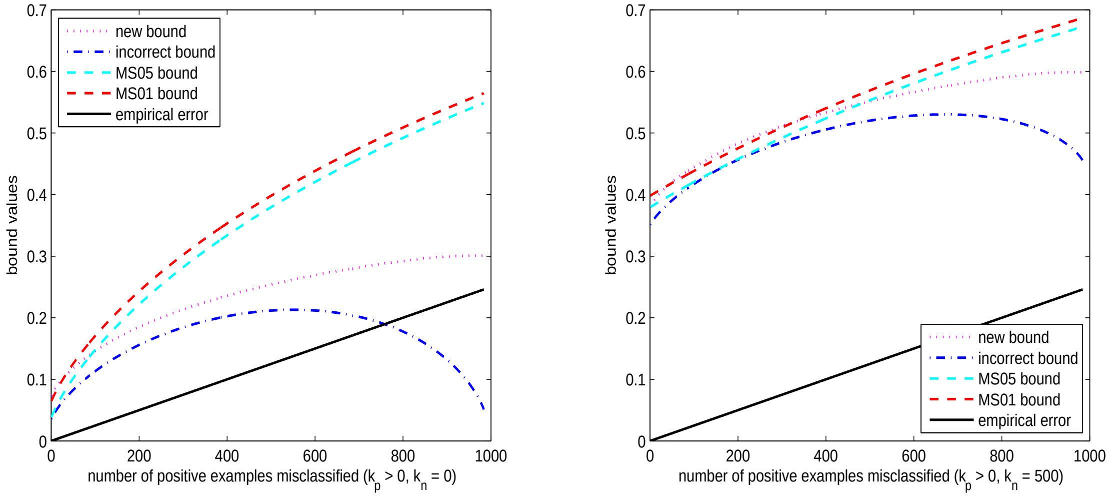

Figure 2: Bound values for the SCM when $m_p = 1000, m_n = 3000, c_p = 5, c_n = 5, b = 10$.

6. Conclusion

We have observed that the SCM loss bound proposed by Marchand and Shawe-Taylor (2002) is incorrect and, in fact, becomes erroneous in the limit where the number of errors on the positive training examples approaches the total number of positive training examples. We have then proposed a new loss bound, valid for any sample-compression learning algorithm (including the SCM), that depends on the observed fraction of positive examples and on what the classifier achieves on them. This new bound captures the spirit of Marchand and Shawe-Taylor (2002) with very similar tightness in the regimes in which the bound could hold. This is shown in numerical comparisons of the loss bound proposed in this paper with all of the earlier bounds that can be applied to the SCM.

As mentioned above, an advantage of the bound is its ability to take into account the observed number of positive examples in the training set in order to arrive at tighter estimates. It also has the advantage of being applicable in cases where the loss function is asymmetrical for type I and type II errors, a situation that is not uncommon in practical applications.

The tightness of the bounds derived for the set covering machine make it tempting to use them to perform model selection as well as to consider integrating them more closely into the workings of the algorithm. Both of these directions are the subject of ongoing research.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This work was supported by NSERC Discovery grants 262067 and 122405 and, in part, by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

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

John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005.

Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. Technical report, University of California Santa Cruz, Santa Cruz, CA, 1986.

Mario Marchand and John Shawe-Taylor. Learning with the set covering machine. *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 345–352, 2001.

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

Mario Marchand and Marina Sokolova. Learning with decision lists of data-dependent features. *Journal of Machine Learning Reasearch*, 6:427–451, 2005.