A PAC-Bayes Sample Compression Approach to Kernel Methods

Pascal Germain Pascal.Germain@ift.ulaval.ca
Alexandre Lacoste Alexandre.Lacoste.1@ulaval.ca
François Laviolette Francois.Laviolette@ift.ulaval.ca
Mario Marchand Mario.Marchand@ift.ulaval.ca
Sara Shanian Sara.Shanian@ift.ulaval.ca
Département d'informatique et de génie logiciel, Université Laval, Québec, Canada, G1V 0A6

Abstract

We propose a PAC-Bayes sample compression approach to kernel methods that can accommodate any bounded similarity function and show that the support vector machine (SVM) classifier is a particular case of a more general class of data-dependent classifiers known as majority votes of sample-compressed classifiers. We provide novel risk bounds for these majority votes and learning algorithms that minimize these bounds.

1. Introduction

Kernel methods such as the support vector machine (SVM) have provided state-of-the-art machine learning algorithms over the last decade. Despite their success, these methods are currently limited by the fact that the similarity function that they use must be a symmetric positive semi-definite kernel. However, for many applications, we would like to be able to use any similarity measure of input examples and not be limited by the fact that the chosen measure should be expressible as an inner product of feature vectors. We therefore propose here a PAC-Bayes sample-compression approach to kernel methods that can accommodate any bounded similarity function. Within the sample-compression framework (Floyd & Warmuth, 1995; Laviolette & Marchand, 2007) each sample-compressed classifier is partly represented by a subsequence of the training data, called the compression sequence. We show here that the SVM classifier is actually a particular case of a (weighted) majority vote of sample-compressed classifiers where the compression sequence of each classifier consists of at most a single training example. Inspired by the work of Germain et al. (2009) on general loss bounds for stochastic classifiers, we propose two different PAC-Bayes risk bounds for majority votes of sample-compressed classifiers which are valid for any similarity measure of input examples. Consequently, the proposed bounds also apply to the class of linear classifiers of similarity-based features that were studied by Chen et al. (2009a). Indeed, for the class of indefinite similarity measures, their risk bound is trivial (and useless) in the limit where each training example is used for a prototype. In contrast, the risk bounds presented here do not suffer from such a limitation.

For each proposed risk bound, we provide a learning algorithm that minimizes it. One of the PAC-Bayes risk bound depends on the KL divergence between the prior and the posterior over the set of sample-compressed classifiers and, consequently, the corresponding bound-minimizing learning algorithm is KL-regularized. The other PAC-Bayes risk bound has the unusual property of having no KL divergence when the posterior is *aligned* with the prior in some precise way defined below. Consequently, to minimize this risk upper bound, we only need to minimize the proposed empirical loss under the constraint that the posterior is kept aligned with the prior. When a positive semi-definite (PSD) kernel is used, our experiments indicate that the proposed algorithms are very competitive with the SVM. Good empirical results are also obtained when the proposed algorithms are used with a non-PSD kernel. Finally, the proposed algorithms are also competitive with the best similarity-based learning algorithms proposed by Chen et al. (2009a).

2. PAC-Bayesian Sample Compression

We consider binary classification problems where an example $z = (x, y) \in \mathcal{Z}$ is an input-output pair where

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

$x \in \mathcal{X}$ and $y \in \mathcal{Y} = \{-1, +1\}$. We adopt the PAC setting where each example z is drawn according to a fixed, but unknown, probability distribution D on $\mathcal{Z}$. In the *sample compression setting*, learning algorithms have access to a data-dependent set of classifiers, that we refer to as *sc-classifiers*. Given a training sequence $S = \langle z_1, .., z_m \rangle$, each sc-classifier is described by a subsequence $S_{\mathbf{i}}$ of S called the *compression sequence*, and a *message* μ which represents the additional information needed to obtain a classifier from $S_{\mathbf{i}}$. The compression sequence $S_{\mathbf{i}}$ is defined by the following vector $\mathbf{i}$ of indices

$$\mathbf{i} \stackrel{\text{def}}{=} \langle i_1, i_2, \ldots, i_{|\mathbf{i}|} \rangle ,$$

with $1 \leq i_1 < i_2 < \ldots < i_{|\mathbf{i}|} \leq m$. The number of indices present in $\mathbf{i}$ is denoted by $|\mathbf{i}|$, and the vector of indices of a sc-classifier h by $\mathbf{i}_h$. The set of all the 2^m possible vectors of indices is denoted by $\mathcal{I}$. The fact that each sc-classifier is described by a compression sequence and a message implies that there exists a *reconstruction function* $\mathcal{R}$ that outputs a classifier

$$h^{\mu}_{S'} \stackrel{\text{def}}{=} \mathcal{R}(S', \mu) \tag{1}$$

when given an arbitrary compression sequence S' and a message μ chosen from the set $\mathcal{M}_{S'}$ of all messages that can be supplied with the compression sequence S'. $\mathcal{M}_{S'}$ must be defined a priori (before observing the training data) for all possible sequences S'. The messages can be strings or values taken from a continuous set. In our case, $\mathcal{M}_{S'}$ will be continuous.

Given a training sequence S, $\mathcal{H}^S$ denotes the set of all sc-classifiers $\mathcal{R}(S_{\mathbf{i}}, \mu)$ such that $\mu \in \mathcal{M}_{S_{\mathbf{i}}}$ and $\mathbf{i} \in \mathcal{I}$. The perceptron learning rule and the SVM are examples where the final classifier can be reconstructed solely from a compression sequence. In contrast, the reconstruction function of the Set Covering Machine (Marchand & Shawe-Taylor, 2002) needs both a compression sequence and a message string.

The risk $R_D(h)$ (or generalization error) and the *empirical risk* $R_S(h)$ on S of a sc-classifier h are defined as

$$R_D(h) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{(x,y)\sim D} I(h(x) \neq y) = \Pr_{(x,y)\sim D} (h(x) \neq y)$$

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

where $I(a) = 1$ if predicate a is true and 0 otherwise, and where $(x, y) \sim S$ means that (x, y) is drawn according to the uniform distribution on S.

Note that both R_D and R_S are defined only within the context of a training sequence S. In the non-sample compressed setting, $mR_S(h)$ is a binomial random variable of parameters $(m, R_D(h))$. In our setting this is no longer the case because the risk can then be biased by the elements of S that are in the compression sequence. However, if $a_h \stackrel{\text{def}}{=} \sum_{(x,y)\in S_{\mathbf{i}_h}} I(h(x \neq y)$, then $mR_S(h) - a_h$ is a binomial random variable with parameters $(m - |\mathbf{i}|, R_D(h))$. As mentioned in Laviolette & Marchand (2007), the empirical risk of a sc-classifier is usually computed on $S \setminus S_{\mathbf{i}}$. In order to obtain risk bounds having simpler statements, we decide not to follow this strategy and, therefore, deal with this bias directly in the proposed theory.

After observing the training sequence S, the task of the learner is to choose a *posterior* distribution Q over $\mathcal{H}^S$ such that the Q-weighted majority vote classifier B_Q will have the smallest possible risk. On any input example x, the output $B_Q(x)$ of the majority vote classifier (also called the *Bayes classifier*) is given by

$$B_Q(x) \stackrel{\text{def}}{=} \text{sgn}\left[\mathop{\mathbf{E}}_{h\sim Q} h(x) \right], \tag{2}$$

where $\text{sgn}(s) = +1$ if $s > 0$ and $\text{sgn}(s) = -1$ otherwise.

Given a training sequence S, we denote by $Q_{\mathcal{I}}(\mathbf{i})$, the probability that a compression sequence $S_{\mathbf{i}}$ is chosen by Q, and by $Q_{S_{\mathbf{i}}}(\mu)$, the probability distribution of choosing μ given $S_{\mathbf{i}}$. Consequently,

$$Q_{\mathcal{I}}(\mathbf{i}) \stackrel{\text{def}}{=} \int_{\mu\in\mathcal{M}(S_{\mathbf{i}})} Q(h^{\mu}_{S_{\mathbf{i}}})\, d\mu \quad \text{and} \quad Q_{S_{\mathbf{i}}}(\mu) \stackrel{\text{def}}{=} Q(h^{\mu}_{S_{\mathbf{i}}} \mid S_{\mathbf{i}}) .$$

Priors on sc-classifiers. In PAC-Bayes theory, risk bounds are obtained by comparing a *posterior* distribution Q on $\mathcal{H}^S$ to a *prior* defined before observing the training sequence S. Therefore, in standard PAC-Bayes bounds (McAllester, 2003; Seeger, 2002), the prior is independent of S. In our setting, this seems problematic since sc-classifiers are defined upon S. To overcome this difficulty, we follow Laviolette & Marchand (2007) and define a *prior* $\mathcal{P}$ as a couple $\left(P_{\mathcal{I}}, (P_{S'})_{S'\in\mathcal{Z}^j, j\leq m}\right)$, where $P_{\mathcal{I}}$ is a distribution on $\mathcal{I}$, and, for every possible compression sequence S', $P_{S'}$ is a distribution on $\mathcal{M}_{S'}$. Given a training sequence S, P denotes the distribution on $\mathcal{H}^S$ associate with the prior $\mathcal{P}$. Consequently,

$$P(h^{\mu}_{S_{\mathbf{i}}}) \quad = \quad P_{\mathcal{I}}(\mathbf{i})\, P_{S_{\mathbf{i}}}(\mu) . \tag{3}$$

Hence, although $\mathcal{P}$ (and thus $P_{\mathcal{I}}$) is defined without reference to any specific training sequence S, the distribution P on $\mathcal{H}^S$ refers to a specific realization of the prior $\mathcal{P}$ on the observed training sequence S. As a result, the risk bounds of this paper only depend on the observed training sequence and not on some prior distribution over all possible training sequences. The way it is accomplished is detailed in the proof of Claim 1 of Theorem 5 in Germain et al. (2011).

Gibbs classifier. In the usual PAC-Bayes setting, a bound on $R_D(B_Q)$ is indirectly obtained by bounding the risk of an associated stochastic classifier known as the Gibbs classifier. To assign an output label to an input example x, the Gibbs classifier G_Q randomly chooses a classifier h according to Q and uses $h(x)$ for the assigned label. In the sample-compressed PAC-Bayes setting, given a training sequence S, G_Q randomly chooses $\mathbf{i}$ according to $Q_{\mathcal{I}}$, then chooses a message μ according to $Q_{S_\mathbf{i}}$, and then classifies x according to $h^\mu_{S_\mathbf{i}}(x)$. Given a distribution D and a training sequence S generated by D, the true risk $R_D(G_Q)$ and its empirical estimate $R_S(G_Q)$ on S are thus given by

$$R_D(G_Q) \;=\; \mathop{\mathbf{E}}_{h^\mu_{S_\mathbf{i}}\sim Q}\;\mathop{\mathbf{E}}_{(x,y)\sim D} I(h^\mu_{S_\mathbf{i}}(x)\neq y)$$
$$R_S(G_Q) \;=\; \mathop{\mathbf{E}}_{h^\mu_{S_\mathbf{i}}\sim Q}\;\mathop{\mathbf{E}}_{(x,y)\sim S} I(h^\mu_{S_\mathbf{i}}(x)\neq y)\,.$$

Note that whenever B_Q errs on (x,y), at least half of the classifiers, under measure Q err on (x,y). It follows that $R_D(B_Q)\leq 2R_D(G_Q)$. Hence, an upper-bound on $R(G_Q)$ also provides an upper bound on $R(B_Q)$ via this well-known factor-of-two rule. However, we focus in this paper on majority votes of sc-classifiers having a small compression size. In this setting, $\mathcal{H}^S$ consists mostly of "weak" classifiers having large risk $R(h)$. Then, $R_D(G_Q)$ is (almost) always large (near $1/2$) for any Q *even if the majority vote B_Q has very low risk*. Thus, the disparity between $R_D(B_Q)$ and $R_D(G_Q)$ is enormous. Consequently, trying to minimize an upper-bound on $R_D(G_Q)$ should not lead to a majority vote B_Q having low risk. In fact, our experiments of Section 2.4 empirically confirm this to be the case. One way to obtain a more relevant bound on $R_D(B_Q)$ from the PAC-Bayes theory is to use a loss function for stochastic classifiers which is distinct from the zero-one loss used for the deterministic classifiers. In order to obtain a tractable optimization problem, we propose to use a convex loss function of the margin of the Q-convex combination of sc-classifiers where the margin, on example (x,y), is defined as

$$M_Q(x,y) \;\stackrel{\text{def}}{=}\; \mathop{\mathbf{E}}_{h^\mu_{S_\mathbf{i}}\sim Q} yh^\mu_{S_\mathbf{i}}(x)\,. \qquad (4)$$

Note that $R_D(G_Q)=\frac{1}{2}-\frac{1}{2}\,\mathbf{E}_{(\mathbf{x},y)\sim D}M_Q(x,y)$ gives a relation between $R_D(G_Q)$ and $M_Q(x,y)$.

Similarly as in Germain et al. (2009), we only consider losses that upper-bound the zero-one loss of B_Q. Hence, we consider functions $\zeta:[-1,1]\to\mathbb{R}$ of the form

$$\zeta(\alpha)=\sum_{k=0}^{\deg(\zeta)} a_k\,\alpha^k \;\text{ such that }\; \zeta(\alpha)\geq I(-\alpha\leq 0)\,,$$

with $a_k\geq 0$. Then, we will provide PAC-Bayes bounds on the following expected loss

$$\zeta_D^Q \;\stackrel{\text{def}}{=}\; \mathop{\mathbf{E}}_{(x,y)\sim D}\zeta\left(-M_Q(x,y)\right)\,, \qquad (5)$$

based on its empirical (possibly biased) estimate $\zeta_S^Q \stackrel{\text{def}}{=} \mathbf{E}_{(x,y)\sim S}\;\zeta\left(-M_Q(x,y)\right)$. Such a ζ is called a *convex margin loss function* (or a *convex surrogate loss*). Since $\zeta(\alpha)\geq I(-\alpha\leq 0)$ we have

$$\zeta_D^Q \;\geq\; \mathop{\mathbf{E}}_{(x,y)\sim D} I(M_Q(\mathbf{x},y)\leq 0\,) \;\geq\; R_D(B_Q)\,.$$

Thus, ζ_D^Q is always an upper bound of $R_D(B_Q)$. In particular, the factor-of-two rule $R_D(B_Q)\leq 2R_D(G_Q)$ simply corresponds to the case where $a_0=a_1=1$, and $a_j=0$ for all $j>1$, as for these values, $\zeta_D^Q=2R_D(G_Q)$.

The next theorem gives a bound on ζ_D^Q and, consequently, on $R_D(B_Q)$. It can be viewed as a generalization of Theorem 1.2.1 of Catoni (2007) to the sample compression setting and to general margin loss functions. (Proof provided in Germain et al. (2011).)

Theorem 1. *For any D, any family $(\mathcal{H}^S)_{S\in D^m}$ of sets of sc-classifiers of size at most l, any prior $\mathcal{P}$, any $\delta\in(0,1]$, any $C_1>0$, and any margin loss function ζ such that $l\cdot\deg(\zeta)<m$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall Q \text{ on } \mathcal{H}^S\text{:}\\ \zeta_D^Q\leq \zeta(1)[C'-1]+\\ C'\cdot\left(\zeta_S^Q+\frac{2}{m\cdot C_1}[\zeta'(1)\cdot\mathrm{KL}(Q\|P)+\zeta(1)\cdot\ln\frac{1}{\delta}]\right)\end{array}\right)\geq 1-\delta$$

where $\mathrm{KL}(\cdot\|\cdot)$ *is the Kullback-Leibler divergence, and where* $C'=\frac{C_1\cdot\frac{m}{m-l\cdot\deg(\zeta)}}{1-e^{-C_1\cdot\frac{m-l\cdot\deg(\zeta)}{m}}}$.

The bound of Theorem 1 holds for any constant C_1. Even if the bound can be made valid uniformly for k different values of C_1 by replacing δ with δ/k (thanks to the standard union bound argument), this is a disadvantage when one wants to make a bound minimization algorithm out of it because such an algorithm will have to tune this extra hyper-parameter. However, the next subsection shows that restricting Q to be an *aligned posteriors* can provide a PAC-Bayes bound *independent* of the Kullback-Leibler divergence between the posterior and the prior. As a consequence, no constant C_1 will be present in the proposed bound. To our knowledge, this is new in PAC-Bayes theory.

2.1. The Case of Aligned Posteriors

To define the notion of aligned posteriors, we need to consider the boolean complement $-h^\mu_{S'}$ of any sc-classifier $h^\mu_{S'}$. Thus, we now consider that the message sets are of the form $\mathcal{M}_{S'}=\mathcal{M}^1_{S'}\times\{+,-\}$, and that we always have $h^{(\sigma,+)}_{S_\mathbf{i}}=-h^{(\sigma,-)}_{S_\mathbf{i}}\;\forall\sigma\in\mathcal{M}^1_{S_\mathbf{i}}$.

Definition 2. *Given a prior $\mathcal{P}$ and a training sequence S, a posterior Q is said to be* aligned *on P if $Q(h^{(\sigma,+)}_{S_\mathbf{i}}) + Q(h^{(\sigma,-)}_{S_\mathbf{i}}) = P(h^{(\sigma,+)}_{S_\mathbf{i}}) + P(h^{(\sigma,-)}_{S_\mathbf{i}})$ for all $(\mathbf{i},\sigma) \in \mathcal{I} \times \mathcal{M}^1_{S_\mathbf{i}}$.*

Remark 3. *An aligned posterior is completely defined by the values of*

$$w(\mathbf{i},\sigma) \stackrel{def}{=} Q(h^{(\sigma,+)}_{S_\mathbf{i}}) - Q(h^{(\sigma,-)}_{S_\mathbf{i}})\,, \qquad (6)$$

under the constraints $|w(\mathbf{i},\sigma)| \leq P(h^{(\sigma,+)}_{S_\mathbf{i}}) + P(h^{(\sigma,-)}_{S_\mathbf{i}})$ for all $(\mathbf{i},\sigma) \in \mathcal{I} \times \mathcal{M}^1_{S_\mathbf{i}}$. Indeed, it immediately follows that Q can be recovered from $\mathcal{P}$ and w because

$$Q(h^{(\sigma,\pm)}_{S_\mathbf{i}}) = \frac{1}{2}\left(P(h^{(\sigma,+)}_{S_\mathbf{i}}) + P(h^{(\sigma,-)}_{S_\mathbf{i}}) \pm w(\mathbf{i},\sigma)\right). \quad (7)$$

Moreover, given any function $w : \mathcal{I} \times \mathcal{M}^1_{S_\mathbf{i}} \to \mathbb{R}$ satisfying the constraints following Equation (6) the function Q given by Equation (7) is a distribution aligned on P.

The next proposition follows directly from what precedes and points out that there is no loss of expressiveness for majority votes if we restrict ourselves to aligned posteriors.

Proposition 4. *Let $\mathcal{P}$ be a prior, S a training sequence, and Q a distribution on $\mathcal{H}^S$ for which there exists $A > 0$ such that for all $\mathbf{i}$ and σ, $A\,|Q(h^{(\sigma,+)}_{S_\mathbf{i}}) - Q(h^{(\sigma,-)}_{S_\mathbf{i}})| \leq P(h^{(\sigma,+)}_{S_\mathbf{i}}) + P(h^{(\sigma,-)}_{S_\mathbf{i}})$. Let Q' be a distribution aligned on P such that $w'(\mathbf{i},\sigma) = A\,(Q(h^{(\sigma,+)}_{S_\mathbf{i}}) - Q(h^{(\sigma,-)}_{S_\mathbf{i}}))$. Then Q' is Bayes-equivalent to Q (i.e., $B_{Q'}(x) = B_Q(x)\ \forall x \in \mathcal{X}$).*

We now provide a PAC-Bayes bound for aligned posteriors which does not depend on how far is an aligned posterior from the prior.

Theorem 5. *For any D, for any $m \geq 8$, for any family $(\mathcal{H}^S)_{S\in D^m}$ of sets of sc-classifiers of size at most l, for any prior $\mathcal{P}$, for any margin loss function ζ such that $l \cdot \deg(\zeta) < m$, and for any $\delta \in (0,1]$, we have*

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall Q \text{ aligned on } P: \\ \quad \zeta^Q_D \leq \zeta^Q_S + \frac{\zeta(1)}{\sqrt{\frac{1}{2}(m - l\deg\zeta)}}\sqrt{4\,l\deg\zeta + \ln\frac{2\sqrt{m}}{\delta}}\end{array}\right) \geq 1-\delta\,.$$

Proof. There are three main difficulties in this proof that prevents a straightforward reduction to a classical PAC-Bayes proof. The first difficulty is that we want to bound the general loss ζ^Q_D instead of the usual Gibbs' risk $R_D(G_Q)$. We overcome this first difficulty by defining a new family of "abstract" sc-classifiers whose Gibbs' risk is closely related to ζ^Q_D by Equation (9) below. The second difficulty comes from the fact that a part of the training data is used to construct the sc-classifiers and, consequently, the empirical risk ζ^Q_S provides a *biased* estimate of ζ^Q_D. This complicates a lot the evaluation of the expected value of $X_{\overline{P}}$ defined below, which is an essential step in all PAC-Bayes proofs. Claim 1 deals with this problem. Finally, in classical proofs, one can only relate $X_{\overline{P}}$, computed with distribution P, to $X_{\overline{Q}}$, computed with another distribution Q, at the cost of generating an extra term which is the Kullback-Leibler divergence between Q and P. Claim 2, below, shows that if Q is aligned on P, both random variables are the same. To our knowledge, this unexpected result is new in PAC-Bayes theory.

Let S be any training sequence and $d \stackrel{\text{def}}{=} \deg\zeta$. For each $k \in \{0,..,d\}$ and any k-tuple $(h_1,..,h_k)$, let us define $\overline{h}$ as an "abstract" sc-classifier $\overline{h_1..h_k}$ whose "abstract" true risk and empirical risk (resp. the cases where $U = D$ and $U = S$) are defined as

$$\overline{R}_U(\overline{h}) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim U} \frac{1}{2}\left[1 + \prod_{i=1}^{k} -y h_i(x)\right]. \quad (8)$$

For the $k = 0$ case, we have $\overline{R}_U(\overline{h}) = \overline{R}_U(\overline{h_1..h_0}) = 1$.

For each S, let $\overline{\mathcal{H}^S}$ be the set of all such sc-classifiers. For each distribution P and Q on $\mathcal{H}^S$, denote by $\overline{P}$ and $\overline{Q}$, the following distributions on $\overline{\mathcal{H}^S}$:

$$\overline{P}(\overline{h}) \stackrel{\text{def}}{=} \frac{a_k}{\zeta(1)}\prod_{i=1}^{k} P(h_i) \quad \text{and} \quad \overline{Q}(\overline{h}) \stackrel{\text{def}}{=} \frac{a_k}{\zeta(1)}\prod_{i=1}^{k} Q(h_i)\,.$$

Since $\zeta(1) = \sum_{k=0}^{d} a_k$, both $\overline{P}$ and $\overline{Q}$ are probability distributions. Moreover, for $U = D$ and $U = S$, we have

$$\begin{aligned} R_U(G_{\overline{Q}}) &= \mathop{\mathbf{E}}_{\overline{h}\sim\overline{Q}} \overline{R}_U(\overline{h}) \\ &= \sum_{k=0}^{d} \frac{a_k}{\zeta(1)} \mathop{\mathbf{E}}_{h_1\sim Q} \cdots \mathop{\mathbf{E}}_{h_k\sim Q} \mathop{\mathbf{E}}_{(x,y)\sim U} \frac{1}{2}\left[1 + \prod_{i=1}^{k} -y h_i(x)\right] \\ &= \sum_{k=0}^{d} \frac{a_k}{\zeta(1)} \mathop{\mathbf{E}}_{(x,y)\sim U} \frac{1}{2}\left[1 + \prod_{i=1}^{k} \mathop{\mathbf{E}}_{h_i\sim Q} -y h_i(x)\right] \\ &= \frac{1}{2}\left[1 + \frac{1}{\zeta(1)} \mathop{\mathbf{E}}_{(x,y)\sim U} \sum_{k=0}^{d} a_k \left(\mathop{\mathbf{E}}_{h\sim Q} -y h(x)\right)^k\right] \\ &= \frac{1}{2}\left[1 + \frac{1}{\zeta(1)}\,\zeta^Q_U\right]. \end{aligned} \quad (9)$$

Because the compression sequence size of each h_i is at most l, we have $|\mathbf{i}_{\overline{h}}| \leq l \cdot k$ for any $\overline{h} = \overline{h_1..h_k}$.

Similarly as McAllester (2003), we consider the following Laplace transform:

$$X_{\overline{P}} \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{\overline{h}\sim\overline{P}} e^{(m - |\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h}) - \overline{R}_D(\overline{h}))^2}. \quad (10)$$

In Germain et al. (2011), we prove the following:

Claim 1 : $\mathbf{E}_{S\sim D^m} X_{\overline{P}} \leq e^{4ld}\cdot 2\sqrt{m}\,.$

The fact that there is no KL-divergence in the bound is a consequence of the following claim.

Claim 2 : For any posterior Q aligned on P, we have

$$X_{\overline{P}} \;=\; X_{\overline{Q}} \;\stackrel{\text{def}}{=}\; \mathop{\mathbf{E}}_{\overline{h}\sim\overline{Q}} e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}.$$

Proof of Claim 2. For each $k\in\{0,..,d\}$, define $\overline{\mathcal{H}^S_{(k)}}$ as the set of abstract classifiers $\overline{h}$ that are k-tuples $\overline{h_1..h_k}$. Now, for each $\overline{h}\in\overline{\mathcal{H}^S_{(k)}}$ and each $j=0,..,2^k-1$, define $\overline{h}^{[j]} \stackrel{\text{def}}{=} \overline{h_1^{(s_1)}..h_k^{(s_k)}}$, where $s_1s_2..s_k$ is the binary representation of the number j, and where $h^{(0)}=h$ and $h^{(1)}=-h$. For any $\overline{h}\in\overline{\mathcal{H}^S_{(k)}}$, let $\mathcal{G}(\overline{h})$ be the set of all $\overline{h}^{[j]}$ for the different choices of j. Note that, given any two $\overline{h},\overline{h'}\in\overline{\mathcal{H}^S_{(k)}}$, both $\mathcal{G}(\overline{h})$ and $\mathcal{G}(\overline{h'})$ either coincide or are disjoint. They will coincide iff $\overline{h'}=\overline{h}^{[j]}$ for some j, and $\mathbf{i}_{\overline{h}^{[j]}}=\mathbf{i}_{\overline{h}}$. Moreover, if Q is aligned on P:

$$\begin{aligned}
\sum_{j=0}^{2^k-1}\overline{P}(\overline{h}^{[j]}) \;&=\; \frac{a_k}{\zeta(1)}\sum_{\mathbf{s}\in\{0,1\}^k}\prod_{i=1}^{k}P\left(h_i^{(s_i)}\right)\\
&=\; \frac{a_k}{\zeta(1)}\prod_{i=1}^{k}\left[P\left(h_i^{(0)}\right)+P\left(h_i^{(1)}\right)\right]\\
&=\; \frac{a_k}{\zeta(1)}\prod_{i=1}^{k}\left[Q\left(h_i^{(0)}\right)+Q\left(h_i^{(1)}\right)\right]\\
&=\; \frac{a_k}{\zeta(1)}\sum_{\mathbf{s}\in\{0,1\}^k}\prod_{i=1}^{k}Q\left(h_i^{(s_i)}\right) \;=\; \sum_{j=0}^{2^k-1}\overline{Q}(\overline{h}^{[j]}). \quad (11)
\end{aligned}$$

Also, it follows directly from Equation (8) and the property $(q-p)^2=((1-q)-(1-p))^2$ that

$$(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2=(\overline{R}_S(\overline{h}^{[j]})-\overline{R}_D(\overline{h}^{[j]}))^2. \quad (12)$$

From Equations (11) and (12), we now have

$$\begin{aligned}
&\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\overline{P}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}\\
&=\frac{1}{2^k}\sum_{j=0}^{2^k-1}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\overline{P}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}\\
&=\frac{1}{2^k}\sum_{j=0}^{2^k-1}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\overline{P}(\overline{h}^{[j]})e^{(m-|\mathbf{i}_{\overline{h}^{[j]}}|)\cdot 2(\overline{R}_S(\overline{h}^{[j]})-\overline{R}_D(\overline{h}^{[j]}))^2}\\
&=\frac{1}{2^k}\sum_{j=0}^{2^k-1}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\overline{P}(\overline{h}^{[j]})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}\\
&=\frac{1}{2^k}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\sum_{j=0}^{2^k-1}\overline{P}(\overline{h}^{[j]})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}\\
&=\frac{1}{2^k}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\sum_{j=0}^{2^k-1}\overline{Q}(\overline{h}^{[j]})\;e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}\\
&\;\;\vdots\\
&=\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\overline{Q}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}. \quad (13)
\end{aligned}$$

Thus, the following proves Claim 2:

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\overline{h}\sim\overline{P}} e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}\\
&\quad=\sum_{k=0}^{\deg\zeta}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\overline{P}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}\\
&\quad=\sum_{k=0}^{\deg\zeta}\int_{\overline{h}\in\overline{\mathcal{H}^S_{(k)}}}\overline{Q}(\overline{h})e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}\\
&\quad=\mathop{\mathbf{E}}_{\overline{h}\sim\overline{Q}} e^{(m-|\mathbf{i}_{\overline{h}}|)\cdot 2(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2}.
\end{aligned}$$

Now, by Markov's inequality we have

$$\Pr_{S\sim D^m}\left(X_{\overline{P}}\le\frac{1}{\delta}\mathop{\mathbf{E}}_{S\sim D^m}X_{\overline{P}}\right)\ge 1-\delta.$$

Now, by applying the two claims and taking the logarithm on each side of the innermost inequality, we have

$$\Pr_{S\sim D^m}\left(\begin{array}{l}\forall\, Q \text{ aligned on } P:\\ \ln X_{\overline{Q}} \;\le\; \ln\left[\frac{1}{\delta}\,e^{4ld}\cdot 2\sqrt{m}\right]\end{array}\right) \;\ge\; 1-\delta. \quad (14)$$

By using Jensen's inequality on the concavity of $\ln(x)$ and on the convexity of $(q-p)^2$, and by using the fact that $m-|\mathbf{i}_{\overline{h}}|\;\ge\; m-l\cdot d$, we find

$$\begin{aligned}
\ln X_{\overline{Q}} \;&\ge\; \mathop{\mathbf{E}}_{\overline{h}\sim\overline{Q}}(m-|\mathbf{i}_{\overline{h}}|)\cdot 2\cdot(\overline{R}_S(\overline{h})-\overline{R}_D(\overline{h}))^2\\
&\ge\; (m-ld)\cdot 2\cdot\left(\mathop{\mathbf{E}}_{\overline{h}\sim\overline{Q}}\overline{R}_S(\overline{h})-\mathop{\mathbf{E}}_{\overline{h}\sim\overline{Q}}\overline{R}_D(\overline{h})\right)^2.
\end{aligned}$$

The theorem then follows from Equations (9) and (14) and straightforward calculations. □

2.2. Majority Votes of sc-classifiers of Compression Size of at Most One

For the rest of the paper, we specialize ourselves to the case where each sc-classifier has a compression set size of at most one. In that case, each sample compression sequence $S_{\mathbf{i}}$ consists of at most a single training example and, consequently, each possible vector $\mathbf{i}$ has at most only one index (*i.e.*, $|\mathbf{i}|\le 1$). When $|\mathbf{i}|=1$ and its single index points to example (x_i,y_i) of S, we have $\mathbf{i}=\langle i\rangle$ and $S_{\mathbf{i}}=S_{\langle i\rangle}=(x_i,y_i)$. When $|\mathbf{i}|=0$, then $\mathbf{i}=\langle\rangle$ and $S_{\mathbf{i}}=S_{\langle\rangle}=\emptyset$. In this latter case, the two sc-classifiers $h^{(\varepsilon,+)}_{S_{\langle\rangle}}$ and $h^{(\varepsilon,-)}_{S_{\langle\rangle}}$ are constant classifiers so that $h^{(\varepsilon,+)}_{S_{\langle\rangle}}(x)=+1$ and $h^{(\varepsilon,-)}_{S_{\langle\rangle}}(x)=-1$ for all $x\in\mathcal{X}$. Here ε denotes the empty message. Each sc-classifier $h^{(\sigma,s)}_{S_{\langle i\rangle}}$ (that we will define below) of compression size 1 uses a message $(\sigma,s)\in\mathcal{M}^1\times\{-,+\}$ where $\mathcal{M}^1$ is a real interval having a length denoted by $|\mathcal{M}^1|$. Furthermore, we use a *uniform prior* over all the relevant parameters. More precisely, for all $i\in\{1,\ldots,m\}$, and $s\in\{-,+\}$, we have

$$\begin{aligned}
P_{\mathcal{I}}(\langle\rangle) \;&=\; P_{\mathcal{I}}(\langle i\rangle) \;=\; \frac{1}{m+1} \quad;\quad P_{S_{\langle\rangle}}(\varepsilon,s) \;=\; \frac{1}{2}\\
P_{S_{\langle i\rangle}}(\sigma,s) \;&=\; \frac{1}{2|\mathcal{M}^1|}I(\sigma\in\mathcal{M}^1).
\end{aligned}$$

Equation (3) implies that $P(h^{(\varepsilon,s)}_{S_{\langle\rangle}}) = P_{\mathcal{I}}(\langle\rangle)P_{S_{\langle\rangle}}(\varepsilon,s)$ and $P(h^{(\sigma,s)}_{S_{\langle i\rangle}}) = P_{\mathcal{I}}(\langle i\rangle)P_{S_{\langle i\rangle}}(\sigma,s)$. We do have

$$\sum_{s\in\{-,+\}} P(h^{(\varepsilon,s)}_{S_{\langle\rangle}}) + \sum_{i=1}^{m}\sum_{s\in\{-,+\}}\int_{\mathcal{M}^1} d\sigma P(h^{(\sigma,s)}_{S_{\langle i\rangle}}) \;=\; 1\,.$$

In the rest of the paper, we restrict ourselves to what we call *strongly aligned posteriors*, *i.e.*, aligned posteriors such that the function $w(\mathbf{i},\sigma)$ is constant on its second argument. More precisely, for sc-classifiers of compression size zero, we use

$$\begin{aligned} Q(h^{(\varepsilon,s)}_{S_{\langle\rangle}}) &= \frac{1}{2}\left(P(h^{(\varepsilon,+)}_{S_{\langle\rangle}}) + P(h^{(\varepsilon,-)}_{S_{\langle\rangle}}) + s\cdot w(\langle\rangle,\varepsilon)\right) \\ &= \frac{1}{2}\left(\frac{1}{m+1} + s\cdot w_0\right), \end{aligned} \tag{15}$$

where $w_0 \stackrel{\text{def}}{=} w(\langle\rangle,\varepsilon)$ and must satisfy $|w_0| \le \frac{1}{m+1}$.

For sc-classifiers of compression size one, we use

$$\begin{aligned} Q(h^{(\sigma,s)}_{S_{\langle i\rangle}}) &= \frac{1}{2}\left(P(h^{(\sigma,+)}_{S_{\langle i\rangle}}) + P(h^{(\sigma,-)}_{S_{\langle i\rangle}}) + s\cdot w(\langle i\rangle,\sigma)\right) \\ &= \frac{1}{2}\left(\frac{1}{m+1} + s\cdot w_i\right)\frac{1}{|\mathcal{M}^1|}I(\sigma\in\mathcal{M}^1)\,, \end{aligned} \tag{16}$$

where we have defined w_i by the equality $w(\langle i\rangle,\sigma) = w_i\frac{1}{|\mathcal{M}^1|}I(\sigma\in\mathcal{M}^1)$. Thus, we must satisfy $|w_i| \le \frac{1}{m+1}$.

The specialization to strongly aligned posterior might seem too restrictive. However, this setting remains powerful enough to include kernels methods such as the SVM as a *special case*. Indeed, for any such strongly aligned posterior Q, the output of $B_Q(x)$ is

$$B_Q(x) = \operatorname{sgn}\left(w_0 + \sum_{i=1}^{m} w_i k(x_i,x)\right), \tag{17}$$

since $\mathop{\mathbf{E}}_{h\sim Q} h(x)$, in Equation (2), is given by

$$\begin{aligned} &\sum_{s\in\{-,+\}} Q(h^{(\varepsilon,s)}_{S_{\langle\rangle}})h^{(\varepsilon,s)}_{S_{\langle\rangle}}(x) + \sum_{i=1}^{m}\sum_{s\in\{-,+\}}\int_{\mathcal{M}^1} d\sigma Q(h^{(\sigma,s)}_{S_{\langle i\rangle}})h^{(\sigma,s)}_{S_{\langle i\rangle}}(x) \\ &= w_0 + \sum_{i=1}^{m} w_i\frac{1}{|\mathcal{M}^1|}\int_{\mathcal{M}^1} h^{(\sigma,+)}_{S_{\langle i\rangle}}(x)d\sigma = w_0 + \sum_{i=1}^{m} w_i k(x_i,x)\,. \end{aligned}$$

The last equality is obtained whenever $h^{(\sigma,+)}_{S_{\langle i\rangle}}$ satisfies $\int_{\mathcal{M}^1} h^{(\sigma,+)}_{S_{\langle i\rangle}}(x)d\sigma \;=\; |\mathcal{M}^1|k(x_i,x)$. Hence, we choose

$$h^{(\sigma,+)}_{S_{\langle i\rangle}}(x) = \operatorname{sgn}\left(\frac{1}{2}|\mathcal{M}^1|k(x_i,x) > \sigma\right)\;\forall x\in\mathcal{X}\,, \tag{18}$$

for $\sigma\in\mathcal{M}^1 = [-1,+1]$ and $k(x',x)\le 1\;\forall(x',x)\in\mathcal{X}^2$. This last condition implies that k must be bounded by 1. However, note that no other condition need to be satisfied for k. Indeed, k can be any normalized similarity measure; it does not even need to be symmetric.

If we compare the set of majority-vote classifiers described by Equation (17) to the set of SVM classifiers where the output $f_{\text{SVM}}(x)$ for any $x\in\mathcal{X}$ is given by $f_{\text{SVM}}(x) = \operatorname{sgn}\Big(\sum_{i=1}^{m} y_i\alpha_i k(x_i,x) + b\Big)$, we conclude that the latter set forms a strict subset of the former set. Indeed, even if for B_Q we must have $k(x',x)\le 1\;\forall(x',x)\in\mathcal{X}^2$ and $|w_i|\le\frac{1}{m+1}$ for all i, while no such restriction exists for f_{SVM}, we can always multiply b and each α_i by a positive constant such that $f_{\text{SVM}}(x) = B_Q(x)\;\forall x\in\mathcal{X}$. However, k in B_Q can be any similarity measure (possibly not symmetric in its two arguments), while k in f_{SVM} must be a positive semi-definite kernel. Note that Theorems 1 and 5 do apply to this larger class of majority votes of sc-classifiers of compression size of at most one.

Several generalizations from the above are possible. Indeed, for $Q(h^{(\sigma,s)}_{S_{\langle i\rangle}})$, we could consider distributions over σ that vary with i. This would effectively provide a mechanism for adapting the similarity measure to each training example. We could also use sc-classifiers having a compression size larger than one.

2.3. Bound Minimization Learning Algorithms

For the task of finding the posterior Q minimizing an upper bound on ζ^Q_D, note that theorems 1 and 5 indicate that $l\cdot\deg(\zeta)$ should be small for the risk bound to be small. Hence, we consider sc-classifiers of compression sequence size of at most one and margin losses that are quadratic[1], i.e. $\zeta(\alpha) \;=\; (1+\frac{1}{q}\,\alpha)^2$. Hence, we have $\zeta(1) = (1+q^{-1})^2$ and $\zeta'(1) = (2q+2)/q^2$. Algorithm *PBSC-A* finds a strongly aligned Q minimizing ζ^Q_S and, thus, the bound of Theorem 5. Equations (15) and (16) show that any strongly aligned Q is determined by $\mathbf{w}\stackrel{\text{def}}{=}(w_0,w_1,\ldots,w_m)$. When $\mathcal{H}^S$ consists of sc-classifiers of compression sequence size of at most one, as defined in Section 2.2, the margin of Q on (x,y) is $M_Q(x,y) \;=\; y\,[\,w_0 + \sum_{i=1}^{m} w_i k(x_i,x)\,]$. Thus, *PBSC-A* minimizes

$$\begin{aligned} \zeta^Q_S &= \frac{1}{m}\sum_{j=1}^{m}\zeta\left(-M_Q(x_j,y_j)\right)^2 \\ &= \frac{1}{mq^2}\sum_{j=1}^{m}\left(q - y_j\left[w_0 + \sum_{i=1}^{m} w_i k(x_i,x_j)\right]\right)^2, \end{aligned}$$

subject to $|w_i|\le\frac{1}{m+1}$ for $i=1,\ldots,m$. Hence *PBSC-A* solves a least-square problem under an ℓ_∞ norm constraint.

[1] The sc-classifiers being weak, we have seen that linear margin losses are not suitable for model selection.

The objective function to minimize is convex in Q (with continuous first derivatives) and the domain of all strongly aligned posteriors is also convex. The set of solutions is therefore convex and coordinate descent methods, such as the one we have implemented (see Germain et al. (2011) for the details), are thus always guaranteed to converge to a solution.

Similarly, algorithm *PBSC-N* finds the posterior Q that minimizes the bound of theorem 1, without restricting Q to be aligned to the prior. It considers the same set of sc-classifiers of size one defined previously. However, the non-aligned posterior Q is defined by $\mathbf{v} \stackrel{\text{def}}{=} (v_+, v_1, \ldots, v_{2m}, v_-)$ such that :

$$Q(h^{(\varepsilon,+)}_{S_{\langle\rangle}}) = v_+, \quad Q(h^{(\sigma,+)}_{S_{\langle i\rangle}}) = v_i \tfrac{1}{|\mathcal{M}^1|} I(\sigma \in \mathcal{M}^1),$$
$$Q(h^{(\varepsilon,-)}_{S_{\langle\rangle}}) = v_-, \quad Q(h^{(\sigma,-)}_{S_{\langle i\rangle}}) = v_{m+i} \tfrac{1}{|\mathcal{M}^1|} I(\sigma \in \mathcal{M}^1),$$

where $i \in \{1, \ldots, m\}$. For Q to be a distribution, $\mathbf{v}$ must satisfy

$$v \geq 0 \text{ for all } v \in \mathbf{v} \quad \text{and} \quad \sum_{v \in \mathbf{v}} v = 1 \,. \tag{19}$$

Note that the posterior Q minimizing the bound of Theorem 1 is the one that minimizes $C \cdot \zeta^Q_S + \mathrm{KL}(Q\|P)$, subject to the constraints of Equation (19). The parameter $C > 0$ allows to tune the influence of the regularizer. However, in Theorem 1, C is a constant obtained from C_1, $\zeta'(1)$ and m.

To express ζ^Q_S in terms of $\mathbf{v}$, note that the margin $M_Q(x, y)$, on example (x, y), is given by

$$M_Q(x,y) = y \left[(v_+ - v_-) + \sum_{i=1}^{m} (v_i - v_{i+m}) k(x_i, x) \right] .$$

Consequently, we have

$$\zeta^Q_S = \frac{1}{mq^2} \sum_{j=1}^{m} \left(q - y_j \left[v_+ - v_- + \sum_{i=1}^{m} (v_i - v_{i+m}) k(x_i, x_j) \right] \right)^2 .$$

Moreover, the KL-divergence between Q and an uniform prior P is given by

$$\mathrm{KL}(Q\|P) = \mathop{\mathbf{E}}_{h \sim Q} \ln \tfrac{Q(h)}{P(h)} = \ln(2m+2) + \sum_{v \in \mathbf{v}} v \ln v \,.$$

The optimization problem for *PBSC-N* is thus a KL-regularized least-square problem. The objective function to minimize is convex in $\mathbf{v}$ (with continuous first derivatives) and $\mathbf{v}$ lies in a convex domain. A coordinate-pair descent algorithm that works iteratively exchanging weights between two components of $\mathbf{v}$, such as the one we have implemented (see Germain et al. (2011) material for details), is thus assured to converge to a global optimum.

2.4. Empirical Results

We first compare both PBSC algorithms to the popular SVM, to the Regularized Least Squares Classifier (RLSC) [2] , and to an algorithm, denoted here as LINEAR, that just consists at minimizing ζ^Q_S for the linear margin loss function $\zeta(\alpha) = \alpha + 1$. The comparison with the latter is only to point out the need of a non-linear margin loss function. Note that *PBSC-N* has two hyper-parameters (C, q) to tune whereas *PBSC-A* needs only one (q). SVM and RLSC also need to tune only one hyper-parameter (C).

For all algorithms, we used the standard RBF kernel $k_{\mathrm{RBF}}(x, x') = \exp(-\gamma\|x - x'\|)$ and the sigmoid kernel $k_{\mathrm{SIG}}(x, x') = \tanh(s\, x x' + d)$. All hyper-parameters ($C$, q, γ, s, d) were determined by performing 10-fold cross validation on the training data. The experiments were performed on 22 data sets that, except for MNIST, were taken from the UCI repository[3]. Each data set was randomly partitioned into a training set S of size $|S|$ and a testing set T of size $|T|$.

Table 1 shows that, when using the RBF kernel, *PBSC-A* is very competitive with *PBSC-N*, SVM, and RLSC, and outperforms LINEAR. More precisely, *PBSC-A* has a better test risk than SVM on 11 datasets (the opposite occurs 6 times), and *PBSC-A* has a better test risk than RLSC on 12 datasets (the opposite occurs 4 times). Using the sign test methodology of Mendenhall (1983), this yields to a *p-value* of 16.6% for the hypothesis that *PBSC-A* and SVM algorithms are equivalent and a *p-value* of 3.8% for the hypothesis that *PBSC-A* and RLSC algorithms are equivalent. By using the common significance threshold of 5%, we conclude that *PBSC-A* is better than RLSC whereas no such conclusion holds for *PBSC-A* vs SVM. We also have a *p-value* of 10.5% when comparing *PBSC-A* and *PBSC-N*. Note that *PBSC-A* has a better practical value than *PBSC-N* as it requires one hyper-parameter less. Finally, we have a *p-value* of 0.0005% when comparing *PBSC-A* with LINEAR, supporting strongly the choice of the quadratic loss over the linear loss for algorithm design.

Unlike the RBF kernel, the sigmoid kernel is indefinite for certain parameter values. In this case, the standard SVM algorithm might not converge to a solution (like the popular SVM-Light implementation). In our experiments, we used the LIBSVM implementation of Chang & Lin (2001) because it returns a solution even if the kernel is indefinite. It turns out that

[2] RLSC also minimizes a quadratic loss function but with a regularizer different from the one used by PBSC.

[3] Table 1 displays the results for the largest data sets. See Germain et al. (2011) for all the results.

Table 1. Empirical risk measured on the testing set T for the five different algorithms.

Dataset			RBF kernel					Sigmoid kernel	
Name	$\|T\|$	$\|S\|$	**SVM**	**RLSC**	**PBSC-A**	**PBSC-N**	**LINEAR**	**SVM**	**PBSC-A**
Adult	10000	1809	0.158	0.157	**0.156**	0.160	0.193	0.163	**0.157**
BreastC	340	343	**0.038**	**0.038**	0.044	**0.038**	0.144	**0.038**	**0.038**
Letter:AB	1055	500	**0.001**	0.002	**0.001**	**0.001**	0.038	0.009	**0.005**
Letter:DO	1058	500	0.014	0.015	**0.011**	0.012	0.069	**0.022**	0.028
Letter:OQ	1036	500	0.016	**0.011**	0.016	0.014	0.123	**0.018**	0.039
Mnist:0vs8	1916	500	**0.003**	0.009	0.004	0.004	0.031	0.007	**0.003**
Mnist:1vs7	1922	500	0.014	0.008	0.008	**0.007**	0.161	0.012	**0.007**
Mnist:1vs8	1936	500	0.011	**0.010**	**0.010**	0.011	0.292	**0.014**	0.015
Mnist:2vs3	1905	500	0.020	0.022	**0.019**	0.020	0.114	**0.025**	0.031
Mushroom	4062	4062	**0.000**	**0.000**	**0.000**	**0.000**	0.022	**0.000**	0.010
Ringnorm	3700	3700	0.015	0.017	**0.013**	**0.013**	0.103	**0.020**	0.035
Tic-tac-toe	479	479	**0.015**	0.019	0.019	0.052	0.365	**0.023**	0.159
Waveform	4000	4000	0.068	0.067	0.068	**0.066**	0.143	**0.067**	**0.067**

Table 2. Mean and standard deviation (in parentheses) of the empirical risk across 20 partitions.

Dataset	Linear SVM		k-NN		PBSC-A	
Aural Sonar	**0.1425**	(0.694)	0.1825	(0.597)	0.1500	(0.827)
Voting	0.0534	(0.193)	0.0546	(0.174)	**0.0529**	(0.184)
Yeast-5-7	**0.2688**	(0.622)	0.3063	(0.580)	0.2975	(0.668)
Yeast-5-12	**0.1075**	(0.482)	0.1275	(0.439)	0.1088	(0.598)

PBSC-A and LIBSVM are competitive. Indeed, the sign test methodology gives a *p-value* of 50% for the hypothesis that both algorithms are equivalents.

To pursue the exploration with indefinite similarity measures, we ran *PBSC-A* on four binary data sets referenced by Chen et al. (2009a;b) and used the provided similarity measures. We followed the same experimental methodology as Chen et al. (2009a;b) and computed the mean and standard deviation of the empirical risk across 20 test/training standardized partitions. Table 2 shows that *PBSC-A* is competitive with the Linear SVM using similarities as features and is better than the k-Nearest Neighbor using similarities as a measure of distance. Note that Chen et al. (2009b) suggests an algorithm having generally better achievements on these data sets. But these results are obtained by substituting a "surrogate kernel function" to the real similarity function that one wants to use.

References

Catoni, Olivier. *PAC-Bayesian supervised classification: the thermodynamics of statistical learning*. Monograph series of the Institute of Mathematical Statistics, http://arxiv.org/abs/0712.0248, 2007.

Chang, Chih-Chung and Lin, Chih-Jen. *LIBSVM: a library for support vector machines*, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chen, Yihua, Garcia, Eric K., Gupta, Maya R., Rahimi, Ali, and Cazzanti, Luca. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10:747–776, 2009a.

Chen, Yihua, Gupta, Maya R., and Recht, Benjamin. Learning kernels from indefinite similarities. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 145–152, New York, NY, USA, 2009b. ACM.

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

Germain, Pascal, Lacasse, Alexandre, Laviolette, Francois, Marchand, Mario, and Shanian, Sara. From PAC-Bayes bounds to KL regularization. In *Advances in Neural Information Processing Systems 22*, pp. 603–610. 2009.

Germain, Pascal, Lacoste, Alexandre, Laviolette, François, Marchand, Mario, and Shanian, Sara. *A PAC-Bayes Sample compression Approach to Kernel Methods: Supplementary material*, 2011. http://graal.ift.ulaval.ca/publications.php.

Laviolette, François and Marchand, Mario. PAC-Bayes risk bounds for stochastic averages and majority votes of sample-compressed classifiers. *Journal of Machine Learning Research*, 8:1461–1487, 2007.

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

McAllester, David. PAC-Bayesian stochastic model selection. *Machine Learning*, 51:5–21, 2003.

Mendenhall, W. Nonparametric statistics. *Introduction to Probability and Statistics*, 604, 1983.

Seeger, Matthias. PAC-Bayesian generalization bounds for gaussian processes. *Journal of Machine Learning Research*, 3:233–269, 2002.