Journal of Machine Learning Research 2 (2002) 499-526 Submitted 7/01; Published 3/02

Stability and Generalization

Olivier Bousquet BOUSQUET@CMAPX.POLYTECHNIQUE.FR
CMAP, Ecole Polytechnique
F-91128 Palaiseau, FRANCE

André Elisseeff ANDRE.ELISSEEFF@BIOWULF.COM
BIOwulf Technologies
305 Broadway,
New-York, NY 10007

Editor: Dana Ron

Abstract

We define notions of stability for learning algorithms and show how to use these notions to derive generalization error bounds based on the empirical error and the leave-one-out error. The methods we use can be applied in the regression framework as well as in the classification one when the classifier is obtained by thresholding a real-valued function. We study the stability properties of large classes of learning algorithms such as regularization based algorithms. In particular we focus on Hilbert space regularization and Kullback-Leibler regularization. We demonstrate how to apply the results to SVM for regression and classification.

1. Introduction

A key issue in the design of efficient Machine Learning systems is the estimation of the accuracy of learning algorithms. Among the several approaches that have been proposed to this problem, one of the most prominent is based on the theory of uniform convergence of empirical quantities to their mean (see e.g. Vapnik, 1982). This theory provides ways to estimate the *risk* (or generalization error) of a learning system based on an empirical measurement of its accuracy and a measure of its complexity, such as the Vapnik-Chervonenkis (VC) dimension or the fat-shattering dimension (see e.g. Alon et al., 1997).

We explore here a different approach which is based on *sensitivity analysis*. Sensitivity analysis aims at determining how much the variation of the input can influence the output of a system.[1] It has been applied to many areas such as statistics and mathematical programming. In the latter domain, it is often referred to as perturbation analysis (see Bonnans and Shapiro, 1996, for a survey). The motivation for such an analysis is to design robust systems that will not be affected by noise corrupting the inputs.

In this paper, the objects of interest are learning algorithms. They take as input a learning set made of instance-label pairs and output a function that maps instances to the corresponding labels. The sensitivity in that case is thus related to changes of the outcome of the algorithm when the learning set is changed. There are two sources of randomness such algorithms have to cope with: the first one comes from the sampling mechanism used to generate the learning set and the second one is due to noise in the measurements (on the instance and/or label). In contrast to standard approaches to sensitivity analysis, we mainly focus on the sampling randomness and we thus are interested in how changes in the composition of the learning set influence the function produced by the algorithm. The outcome of such an approach is a principled way of getting bounds on the difference between

1. For a qualitative discussion about sensitivity analysis with links to other resources see e.g. `http://sensitivity-analysis.jrc.cec.eu.int/`

empirical and generalization error. These bounds are obtained using powerful statistical tools known as concentration inequalities. The latter are the mathematical device corresponding to the following statement, from Talagrand (1996):

> A random variable that depends (in a "smooth way") on the influence of many independent variables (but not too much on any of them) is essentially constant.

The expression "essentially constant" actually means that the random variable will have, with high probability, a value close to its expected value. We will apply these inequalities to the random variable we are interested in, that is, the difference between an empirical measure of error and the true generalization error. We will see that this variable has either a zero expectation or it has a nice property: the condition under which it concentrates around its expectation implies that its expectation is close to zero. That means that if we impose conditions on the learning system such that the difference between the empirical error and the true generalization error is roughly constant, then this constant is zero. This observation and the existence of concentration inequalities will allow us to state exponential bounds on the generalization error of a stable learning system.

The outline of the paper is as follows: after reviewing previous work in the area of stability analysis of learning algorithms, we introduce three notions of stability (Section 3) and derive bounds on the generalization error of stable learning systems (Section 4). In Section 5, we show that many existing algorithms such as SVM for classification and regression, ridge regression or variants of maximum relative entropy discrimination do satisfy the stability requirements. For each of these algorithms, it is then possible to derive original bounds which have many attractive properties.

Previous work

It has long been known that when trying to estimate an unknown function from data, one needs to find a tradeoff between bias and variance.[2] Indeed, on one hand, it is natural to use the largest model in order to be able to approximate any function, while on the other hand, if the model is too large, then the estimation of the best function in the model will be harder given a restricted amount of data. Several ideas have been proposed to fight against this phenomenon. One of them is to perform estimation in several models of increasing size and then to choose the best estimator based on a complexity penalty (e.g. Structural Risk Minimization). This allows to control the complexity while allowing to use a large model. This technique is somewhat related to regularization procedures that we will study in greater detail in subsequent sections. Another idea is to use statistical procedures to reduce the variance without altering the bias. One such technique is the bagging approach of Breiman (1996a) which consists in averaging several estimators built from random subsamples of the data.

Although it is generally accepted that having a low variance (or a high stability in our terminology) is a desirable property for a learning algorithm, there are few quantitative results relating the generalization error to the stability of the algorithm with respect to changes in the training set. The first such results were obtained by Devroye, Rogers and Wagner in the seventies (see Rogers and Wagner, 1978, Devroye and Wagner, 1979a,b). Rogers and Wagner (1978) first showed that the variance of the leave-one-out error can be upper bounded by what Kearns and Ron (1999) later called *hypothesis stability*. This quantity measures how much the function learned by the algorithm will change when one point in the training set is removed. The main distinctive feature of their approach is that, unlike VC-theory based approaches where the only property of the algorithm that matters is the size of the space to be searched, it focuses on how the algorithm searches the space. This explains why it has been successfully applied to the k-Nearest Neighbors algorithm (k-NN) whose search space is known to have an infinite VC-dimension. Indeed, results from VC-theory

2. We deliberately do not provide a precise definition of bias and variance and resort to common intuition about these notions. In broad terms, the bias is the best error that can be achieved and the variance is the difference between the typical error and the best error.

would not be of any help in that case since they are meaningful when the learning algorithm performs minimization of the empirical error in the full function space. However, the k-NN algorithm is very stable because of its 'locality'. This allowed Rogers and Wagner to get an upper bound on the difference between the leave-one-out error and the generalization error of such a classifier. These results were later extended to obtain bounds on the generalization error of k-local rules in Devroye and Wagner (1979a), and of potential rules in Devroye and Wagner (1979b).

In the early nineties, concentration inequalities became popular in the probabilistic analysis of algorithms, due to the work of McDiarmid (1989) and started to be used as tools to derive generalization bounds for learning algorithms by Devroye (1991). Building on this technique, Lugosi and Pawlak (1994) obtained new bounds for the k-NN, kernel rules and histogram rules. These bounds used "smoothed estimates" of the error which estimate the posterior probability of error instead of simply counting the errors. This smoothing is very much related to the use of real-valued classifiers and we will see that it is at the heart of the applicability of stability analysis to classification algorithms. A comprehensive account of the application of McDiarmid's inequality to obtain bounds for the leave-one-out error or the smoothed error of local classifiers can be found in Devroye et al. (1996).

Independently from this theoretical analysis, practical methods have been developed to deal with instability of learning algorithms. In particular, Breiman (1996a,b) introduced the Bagging technique which is presented as a method to combine single classifiers in such a way that the variance of the overall combination is decreased. However, there is no theoretical guarantee that this variance reduction will bring an improvement on the generalization error.

Finally, a more recent work has shown an interesting connection between stability and VC-theory. Kearns and Ron (1999) derived what they called sanity-check bounds. In particular, they proved that an algorithm having a search space of finite VC-dimension, is stable in the sense that its stability (in a sense to be defined later) is bounded by its VC-dimension. Thus using the stability as a complexity measure does not give worse bounds than using the VC-dimension.

The work presented here follows and extends the stability approach of Lugosi and Pawlak (1994) in that we derive exponential upper bounds on the generalization error based on notions of stability. It is based on earlier results presented in Bousquet and Elisseeff (2001). We consider both the leave-one-out error and the empirical error as possible estimates of the generalization error. We prove stability bounds for a large class of algorithms which includes the Support Vector Machines, both in the regression and in the classification cases. Also we generalize some earlier results from Devroye and Wagner.

2. Preliminaries

We first introduce some notation and then the main tools we will use to derive inequalities.

2.1 Notations

$\mathcal{X}$ and $\mathcal{Y} \subset \mathbb{R}$ being respectively an input and an output space, we consider a training set

$$S = \{z_1 = (x_1, y_1), .., z_m = (x_m, y_m)\},$$

of size m in $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ drawn i.i.d. from an unknown distribution D. A learning algorithm is a function A from $\mathcal{Z}^m$ into $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ which maps a learning set S onto a function A_S from $\mathcal{X}$ to $\mathcal{Y}$. To avoid complex notation, we consider only deterministic algorithms. It is also assumed that the algorithm A is symmetric with respect to S, *i.e.* it does not depend on the order of the elements in the training set. Furthermore, we assume that all functions are measurable and all sets are countable which does not limit the interest of the results presented here.

Given a training set S of size m, we will build, for all $i = 1. \ldots, m$, modified training sets as follows:

- By *removing* the i-th element

$$S^{\setminus i} = \{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_m\} .$$

- By *replacing* the i-th element

$$S^i = \{z_1, \ldots, z_{i-1}, z_i', z_{i+1}, \ldots, z_m\} .$$

  where the replacement example z_i' is assumed to be drawn from D and is independent from S.

Unless they are clear from context, the random variables over which we take probabilities and expectation will be specified in subscript. We thus introduce the notation $\mathbb{P}_S[.]$ and $\mathbb{E}_S[.]$ to denote respectively the probability and the expectation with respect to the random draw of the sample S of size m (drawn according to D^m). Similarly, $\mathbb{P}_z[.]$ and $\mathbb{E}_z[.]$ will denote the probability and expectation when z is sampled according to D.

In order to measure the accuracy of the predictions of the algorithm, we will use a *cost function* $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$. The *loss* of an hypothesis f with respect to an example $z = (x, y)$ is then defined as

$$\ell(f, z) = c(f(x), y) .$$

We will consider several measures of the performance of an algorithm. The main quantity we are interested in is the *risk* or *generalization error*. This is a random variable depending on the training set S and it is defined as

$$R(A, S) = \mathbb{E}_z\left[\ell(A_S, z)\right] .$$

Unfortunately, R cannot be computed since D is unknown. We thus have to estimate it from the available data S. We will consider several estimators for this quantity.

The simplest estimator is the so-called *empirical error* (also known as *resubstitution estimate*) defined as

$$R_{emp}(A, S) = \frac{1}{m} \sum_{i=1}^{m} \ell(A_S, z_i) .$$

Another classical estimator is the *leave-one-out error* (also known as *deleted estimate*) defined as

$$R_{loo}(A, S) = \frac{1}{m} \sum_{i=1}^{m} \ell(A_{S^{\setminus i}}, z_i) .$$

When the algorithm is clear from context, we will simply write $R(S)$, $R_{emp}(S)$ and $R_{loo}(S)$. We will often simplify further the notations when the training sample is clear from context. In particular, we will use the following shorthand notations $R \equiv R(A, S)$, $R_{emp} \equiv R_{emp}(A, S)$, and $R_{loo} \equiv R_{loo}(A, S)$.

2.2 Main Tools

The study we describe here intends to bound the difference between empirical and generalization error for specific algorithms. For any $\epsilon > 0$ our goal is to bound the term

$$\mathbb{P}_S\left[|R_{emp}(A, S) - R(A, S)| > \epsilon\right] , \tag{1}$$

which differs from what is usually studied in learning theory

$$\mathbb{P}_S\left[\sup_{f \in \mathcal{F}} |R_{emp}(f) - R(f)| > \epsilon\right] . \tag{2}$$

Indeed, we do not want to have a bound that holds uniformly over the whole space of possible functions since we are interested in algorithms that may not explore it. Moreover we may not even have a way to describe this space and assess its size. This explains why we want to focus on (1).

Our approach is based on inequalities that relate moments of multi-dimensional random functions to their first order finite differences. The first one is due to Steele (1986) and provides bounds for the variance. The second one is a version of Azuma's inequality due to McDiarmid (1989) and provides exponential bounds but its assumptions are more restrictive.

Theorem 1 (Steele, 1986) *Let S and S^i defined as above, let $F : \mathcal{Z}^m \to R$ be any measurable function, then*

$$\mathbb{E}_S \left[(F(S) - \mathbb{E}_S [F(S)])^2 \right] \leq \frac{1}{2} \sum_{i=1}^{m} \mathbb{E}_{S,z_i'} \left[\left(F(S) - F(S^i) \right)^2 \right]$$

Theorem 2 (McDiarmid, 1989) *Let S and S^i defined as above, let $F : \mathcal{Z}^m \to R$ be any measurable function for which there exists constants c_i $(i = 1, \ldots, m)$ such that*

$$\sup_{S \in \mathcal{Z}^m, z_i' \in \mathcal{Z}} \left| F(S) - F(S^i) \right| \leq c_i \,,$$

then

$$\mathbb{P}_S \left[F(S) - \mathbb{E}_S [F(S)] \geq \epsilon \right] \leq e^{-2\epsilon^2 / \sum_{i=1}^{n} c_i^2} \,.$$

3. Defining the Stability of a Learning Algorithm

There are many ways to define and quantify the stability of a learning algorithm. The natural way of making such a definition is to start from the goal: we want to get bounds on the generalization error of specific learning algorithm and we want these bounds to be tight when the algorithm satisfies the stability criterion.

As one may expect, the more restrictive a stability criterion is, the tighter the corresponding bound will be.

In the learning model we consider, the randomness comes from the sampling of the training set. We will thus consider stability with respect to changes in the training set. Moreover, we need an easy to check criterion so that we will consider only restricted changes such as the removal or the replacement of one single example in the training set.

Although not explicitly mentioned in their work, the first such notion was used by Devroye and Wagner (1979a) in order to get bounds on the variance of the error of *local* learning algorithms. Later, Kearns and Ron (1999) stated it as a definition and gave it a name. We give here a slightly modified version of Kearns and Ron's definition that suits our needs.

Definition 3 (Hypothesis Stability) *An algorithm A has* hypothesis stability *β with respect to the loss function ℓ if the following holds*

$$\forall i \in \{1, \ldots, m\}, \; \mathbb{E}_{S,z} \left[|\ell(A_S, z) - \ell(A_{S^{\setminus i}}, z)| \right] \leq \beta \,. \tag{3}$$

Note that this is the L_1 norm with respect to D, so that we can rewrite the above as

$$\mathbb{E}_S \left[\|\ell(A_S, .) - \ell(A_{S^{\setminus i}}, .)\|_1 \right] \leq \beta$$

We will also use a variant of the above definition in which instead of measuring the average change, we measure the change at one of the training points.

Definition 4 (Pointwise Hypothesis Stability) *An algorithm A has* pointwise hypothesis stability *β with respect to the loss function ℓ if the following holds*

$$\forall i \in \{1, \ldots, m\}, \; \mathbb{E}_S \left[|\ell(A_S, z_i) - \ell(A_{S^{\setminus i}}, z_i)| \right] \leq \beta \,. \tag{4}$$

Another, weaker notion of stability was introduced by Kearns and Ron. It consists of measuring the change in the expected error of the algorithm instead of the average pointwise change.

Definition 5 (Error Stability) *An algorithm A has* error stability β *with respect to the loss function ℓ if the following holds*

$$\forall S \in \mathcal{Z}^m,\ \forall i \in \{1,\ldots,m\},\ |\mathbb{E}_z\left[\ell(A_S,z)\right] - \mathbb{E}_z\left[\ell(A_{S^{\setminus i}},z)\right]| \leq \beta\,, \tag{5}$$

which can also be written

$$\forall S \in \mathcal{Z}^m,\ \forall i \in \{1,\ldots,m\},\ |R(S) - R^{\setminus i}(S)| \leq \beta\,. \tag{6}$$

Finally, we introduce a stronger notion of stability which will allow to get tight bounds. Moreover we will show that it can be applied to large classes of algorithms.

Definition 6 (Uniform Stability) *An algorithm A has* uniform stability β *with respect to the loss function ℓ if the following holds*

$$\forall S \in \mathcal{Z}^m,\ \forall i \in \{1,\ldots,m\},\ \|\ell(A_S,.) - \ell(A_{S^{\setminus i}},.)\|_\infty \leq \beta\,. \tag{7}$$

Notice that (3) implies (5) and (7) implies (3) so that uniform stability is the strongest notion.

Considered as a function of m, the term β will sometimes be denoted by β_m. We will say that an algorithm is *stable* when the value of β_m decreases as $\frac{1}{m}$. An algorithm with uniform stability β has also the following property:

$$\forall S,\ \forall z_i',\ |\ell(A_S,z) - \ell(A_{S^i},z)| \leq |\ell(A_S,z) - \ell(A_{S^{\setminus i}},z))| + |\ell(A_{S^i},z) - \ell(A_{S^{\setminus i}},z)| \leq 2\beta\,.$$

In other words, stability with respect to the exclusion of one point implies stability with respect to changes of one point.

We will assume further that as a function of the sample size, the stability is non-increasing. This will be the case in all our examples. This assumption is not restrictive since its only purpose is to simplify the statement of the theorems (we will always upper bound β_{m-1} by β_m).

4. Generalization Bounds for Stable Learning Algorithms

We start this section by introducing a useful lemma about the bias of the estimators we study.

Lemma 7 *For any symmetric learning algorithm A, we have $\forall i \in \{1,..,m\}$:*

$$\mathbb{E}_S\left[R(A,S) - R_{emp}(A,S)\right] = \mathbb{E}_{S,z_i'}\left[\ell(A_S,z_i') - \ell(A_{S^i},z_i')\right]\,,$$

and

$$\mathbb{E}_S\left[R(A,S^{\setminus i}) - R_{loo}(A,S)\right] = 0\,,$$

and

$$\mathbb{E}_S\left[R(A,S) - R_{loo}(A,S)\right] = \mathbb{E}_{S,z}\left[\ell(A_S,z) - \ell(A_{S^{\setminus i}},z)\right]\,,$$

Proof For the first equality, we just need to compute the expectation of $R_{emp}(A,S)$. We have

$$\mathbb{E}_S\left[R_{emp}(S)\right] = \frac{1}{m}\sum_{j=1}^m \mathbb{E}_S\left[\ell(A_S,z_j)\right] = \frac{1}{m}\sum_{j=1}^m \mathbb{E}_{S,z_i'}\left[\ell(A_S,z_j)\right]\,,$$

and renaming z_j as z_i' we get, $\forall i \in \{1,..,m\}$

$$\mathbb{E}_S\left[R_{emp}(S)\right] = \mathbb{E}_{S,z_i'}\left[\ell(A_{S^i},z_i')\right]\,,$$

by the i.i.d. and the symmetry assumptions. This proves the first equality. Similarly we have

$$\mathbb{E}_S\left[R_{loo}(S)\right] = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_S\left[\ell(A_{S^{\backslash i}}, z_i)\right] = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{S,z}\left[\ell(A_{S^{\backslash i}}, z)\right] ,$$

from which we deduce the second and third equalities. ■

Remark 8 *We notice from the above lemma, comparing the first and last equalities, that the empirical error and the leave-one-out error differ from the true error in a similar way. It is usually accepted that the empirical error is very much optimistically biased while the leave-one-out error is almost unbiased (due to the second equation of the lemma). However, we will see that for the particular algorithms we have in mind (which display high stability), the two estimators are very close to each other. The similarity of the bounds we will derive for both estimators will be striking. This can be explained intuitively by the fact that we are considering algorithms that do not directly minimize the empirical error but rather a regularized version of it, so that the bias in the empirical error will be reduced.*

4.1 Polynomial Bounds with Hypothesis Stability

In this section we generalize a lemma from Devroye and Wagner (1979b). Their approach consists in bounding the second order moment of the estimators with the hypothesis stability of the algorithm. For this purpose, one could simply use Theorem 1. However this theorem gives a bound on the variance and we need here the second order moment of the difference between the error (leave-one-out or empirical) and the generalization error. It turns out that a direct study of this quantity leads to better constants than the use of Theorem 1.

Lemma 9 *For any learning algorithm A and loss function ℓ such that $0 \le c(y, y') \le M$ we have for any $i, j \in \{1, \ldots, m\}$, $i \neq j$ for the empirical error,*

$$\mathbb{E}_S\left[(R - R_{emp})^2\right] \le \frac{M^2}{2m} + 3M\mathbb{E}_{S,z_i'}\left[|\ell(A_S, z_i) - \ell(A_{S^i}, z_i)|\right] , \tag{8}$$

and

$$\begin{aligned}\mathbb{E}_S\left[(R - R_{emp})^2\right] \quad \le \quad & \frac{M^2}{2m} + M\mathbb{E}_{S,z_i',z}\left[|\ell(A_S, z) - \ell(A_{S^i}, z)|\right] \\ & + M\mathbb{E}_{S,z_i'}\left[|\ell(A_S, z_j) - \ell(A_{S^i}, z_j)|\right] \\ & + M\mathbb{E}_{S,z_i'}\left[|\ell(A_S, z_i) - \ell(A_{S^i}, z_i)|\right] ,\end{aligned}$$

and for the leave-one-out error,

$$\mathbb{E}_S\left[(R - R_{loo})^2\right] \le \frac{M^2}{2m} + 3M\mathbb{E}_{S,z}\left[|\ell(A_S, z) - \ell(A_{S^{\backslash i}}, z)|\right] , \tag{9}$$

and

$$\mathbb{E}_S\left[(R - R_{loo})^2\right] \le \frac{M^2}{2m} + 2M\mathbb{E}_{S,z_i',z}\left[|\ell(A_S, z) - \ell(A_{S^i}, z)| + |\ell(A_S, z) - \ell(A_{S^{\backslash i}}, z)|\right] . \tag{10}$$

The proof of this lemma is given in the appendix.

Remark 10 *Notice that Devroye and Wagner's work focused on the leave-one-out estimator and on classification. We extend it to regression and to the empirical estimator, which they treated with the following easy-to-prove inequality*

$$\mathbb{E}_S\left[(R - R_{emp})^2\right] \le 2\mathbb{E}_S\left[(R - R_{loo})^2\right] + 2M\mathbb{E}_S\left[|\ell(A_S, z_i) - \ell(A_{S^{\backslash i}}, z_i)|\right] ,$$

which gives a similar result but with worse constants.

Let's try to explain the various quantities that appear in the upper bounds of the above lemma. We notice that the term $\frac{M^2}{2m}$ is always present and it cannot be avoided even for a very stable algorithm and somehow corresponds to the bias of the estimator. In Inequality (8), the expectation in the right-hand side corresponds to the following situation: starting from training set S we measure the error at point $z_i \in S$, then we replace $z_i \in S$ by z_i' and we again measure the error at z_i which is no longer in the training set. Then, in the second inequality of Lemma 9 several different quantities appear. They all correspond to comparing the algorithm trained on S and on S^i (where z_i is replaced by z_i') but the comparison point differs: it is either z, a point which is not part of the training set, or z_j, a point of the training set different from z_i or finally z_i.

For the leave-one-out error, in (9) we consider the average difference in error when trained on S and on $S^{\backslash i}$ (where z_i has been removed) and in (10), the first expectation in the right hand side corresponds to the average difference in error when one point is changed while the second one is the average difference in error when one point is removed.

All these quantities capture a certain aspect of the stability of the algorithm. In order to use the lemma, we need to bound them for specific algorithms. Instead of using all these different quantities, we will rather focus on the few notions of stability we introduced and see how they are related. We will see later how they can be computed (or upper bounded) in particular cases.

Now that we have a bound on the expected squared deviation of the estimator to the true error, the next step is to use Chebyshev's inequality in order to get a bound which holds with high probability on the deviation.

Theorem 11 *For any learning algorithm A with hypothesis stability β_1 and pointwise hypothesis stability β_2 with respect to a loss function ℓ such that $0 \leq c(y, y') \leq M$, we have with probability $1-\delta$,*

$$R(A,S) \leq R_{emp}(A,S) + \sqrt{\frac{M^2 + 12Mm\beta_2}{2m\delta}}\,,$$

and

$$R(A,S) \leq R_{loo}(A,S) + \sqrt{\frac{M^2 + 6Mm\beta_1}{2m\delta}}\,.$$

Proof First, notice that for all S and all z,

$$\left|\ell(A_S,z) - \ell(A_{S^i},z)\right| \leq \left|\ell(A_S,z) - \ell(A_{S^{\backslash i}},z)\right| + \left|\ell(A_{S^{\backslash i}},z) - \ell(A_{S^i},z)\right|,$$

so that we get

$$\mathbb{E}_{S,z_i'}\left[\left|\ell(A_S,z_i) - \ell(A_{S^i},z_i)\right|\right] \leq \mathbb{E}_S\left[\left|\ell(A_S,z_i) - \ell(A_{S^{\backslash i}},z_i)\right|\right] + \mathbb{E}_{S,z_i'}\left[\left|\ell(A_{S^{\backslash i}},z_i) - \ell(A_{S^i},z_i)\right|\right] \leq 2\beta_2\,.$$

We thus get by (8)

$$\mathbb{E}_S\left[(R - R_{emp})^2\right] \leq \frac{M^2}{2m} + 6M\beta_2\,.$$

Also, we have by (10)

$$\mathbb{E}_S\left[(R - R_{loo})^2\right] \leq \frac{M^2}{2m} + 3M\beta_1\,.$$

Now, recall that Chebyshev's inequality gives for a random variable X

$$\mathbb{P}\left[X \geq \epsilon\right] \leq \frac{\mathbb{E}\left[X^2\right]}{\epsilon^2}\,,$$

which in turn gives that for all $\delta > 0$, with probability at least $1-\delta$,

$$X \leq \sqrt{\frac{\mathbb{E}\left[X^2\right]}{\delta}}\,.$$

Applying this to $R - R_{emp}$ and $R - R_{loo}$ respectively give the result. ■

As pointed out earlier, there is a striking similarity between the above bounds which seems to support the fact that for a stable algorithm, the two estimators that we are considering have a closely related behavior.

In the next section we will see how to use the exponential inequality of Theorem 2 to get better bounds.

4.2 Exponential Bounds with Uniform Stability

Devroye and Wagner (1979a) first proved exponential bounds for k-local algorithms. However, the question of whether their technique can be extended to more general classes of algorithms is a topic for further research.

In Devroye et al. (1996) another, more general technique is introduced which relies on concentration inequalities. Inspired by this approach, we will derive exponential bounds for algorithms based on their uniform stability.

We will study separately the regression and the classification cases for reasons that will be made clear.

4.2.1 Regression Case

A stable algorithm has the property that removing one element in its learning set does not change much of its outcome. As a consequence, the difference between empirical and generalization error, if thought as a random variable, should have a small variance. If its expectation is small, stable algorithms should then be good candidates for their empirical error to be close to their generalization error. This assertion is formulated in the following theorem:

Theorem 12 *Let A be an algorithm with uniform stability β with respect to a loss function ℓ such that $0 \leq \ell(A_S, z) \leq M$, for all $z \in \mathcal{Z}$ and all sets S. Then, for any $m \geq 1$, and any $\delta \in (0,1)$, the following bounds hold (separately) with probability at least $1-\delta$ over the random draw of the sample S,*

$$R \leq R_{emp} + 2\beta + (4m\beta + M)\sqrt{\frac{\ln 1/\delta}{2m}}\,, \tag{11}$$

and

$$R \leq R_{loo} + \beta + (4m\beta + M)\sqrt{\frac{\ln 1/\delta}{2m}}\,. \tag{12}$$

Remark 13 *This theorem gives tight bounds when the stability β scales as $1/m$. We will prove that this is the case for several known algorithms in later sections.*

Proof Let's prove that the conditions of Theorem 2 are verified by the random variables of interest. First we study how these variables change when one training example is removed. We have

$$|R - R^{\setminus i}| \leq \mathbb{E}_z\left[|\ell(A_S, z) - \ell(A_{S^{\setminus i}}, z)|\right] \leq \beta\,, \tag{13}$$

and

$$\begin{aligned} |R_{emp} - R_{emp}^{\setminus i}| &\leq \frac{1}{m}\sum_{j \neq i} |\ell(A_S, z_j) - \ell(A_{S^{\setminus i}}, z_j)| + \frac{1}{m}\,|\ell(A_S, z_i)| \\ &\leq \beta + \frac{M}{m}\,. \end{aligned}$$

Then we upper bound the variation when one training example is changed:

$$|R - R^i| \leq |R - R^{\setminus i}| + |R^{\setminus i} - R^i| \leq 2\beta\,.$$

Similarly we can write

$$|R_{emp} - R^i_{emp}| \leq |R_{emp} - R^{\setminus i}_{emp}| + |R^{\setminus i}_{emp} - R^i_{emp}| \leq 2\beta + 2\frac{M}{m}\,.$$

however, a closer look reveals that the second factor of 2 is not needed. Indeed, we have

$$\begin{aligned}
|R_{emp} - R^i_{emp}| &\leq \frac{1}{m}\sum_{j\neq i} |\ell(A_S, z_j) - \ell(A_{S^i}, z_j)| + \frac{1}{m}\,|\ell(A_S, z_i) - \ell(A_{S^i}, z'_i)| \\
&\leq \frac{1}{m}\sum_{j\neq i} |\ell(A_S, z_j) - \ell(A_{S^{\setminus i}}, z_j)| + \frac{1}{m}\sum_{j\neq i} |\ell(A_{S^{\setminus i}}, z_j) - \ell(A_{S^i}, z_j)| \\
&\quad + \frac{1}{m}\,|\ell(A_S, z_i) - \ell(A_{S^i}, z'_i)| \\
&\leq 2\beta + \frac{M}{m}\,.
\end{aligned}$$

Thus the random variable $R - R_{emp}$ satisfies the conditions of Theorem 2 with $c_i = 4\beta + \frac{M}{m}$.

It thus remains to bound the expectation of this random variable which can be done using Lemma 7 and the β-stability property:

$$\begin{aligned}
\mathbb{E}_S\left[R - R_{emp}\right] &\leq \mathbb{E}_{S,z'_i}\left[|\ell(A_S, z'_i) - \ell(A_{S^i}, z'_i)|\right] \\
&\leq \mathbb{E}_{S,z'_i}\left[|\ell(A_{S^i}, z'_i) - \ell(A_{S^{\setminus i}}, z'_i)|\right] + \mathbb{E}_{S,z'_i}\left[|\ell(A_{S^{\setminus i}}, z'_i) - \ell(A_S, z'_i)|\right] \\
&\leq 2\beta\,.
\end{aligned}$$

Which yields

$$\mathbb{P}_S\left[R - R_{emp} > \epsilon + 2\beta_m\right] \leq \exp\left(-\frac{2m\epsilon^2}{(4m\beta_m + M)^2}\right)\,.$$

Thus, setting the right hand side to δ, we obtain that with probability at least $1-\delta$,

$$R \leq R_{emp} + 2\beta_m + (4m\beta_m + M)\sqrt{\frac{\ln 1/\delta}{2m}}\,,$$

and thus

$$R \leq R_{emp} + 2\beta_m\left(1 + \sqrt{2m\ln 1/\delta}\right) + M\sqrt{\frac{\ln 1/\delta}{2m}}\,,$$

which gives, (11)

For the leave-one-out error, we proceed similarly. We have

$$\begin{aligned}
|R_{loo} - R^{\setminus i}_{loo}| &\leq \frac{1}{m}\sum_{j\neq i} |\ell(A_{S^{\setminus j}}, z_j) - \ell(A_{S^{\setminus i,j}}, z_j)| + \frac{1}{m}\,|\ell(A_{S^{\setminus i}}, z_i)| \\
&\leq \beta_{m-1} + \frac{M}{m}\,,
\end{aligned}$$

and also

$$|R_{loo} - R^i_{loo}| \leq 2\beta_{m-1} + \frac{M}{m} \leq 2\beta_m + \frac{M}{m}\,.$$

So that Theorem 2 can be applied to $R - R_{loo}$ with $c_i = 4\beta_m + \frac{M}{m}$. Then we use Lemma 7 along with (13) to deduce

$$\mathbb{P}_S\left[R - R_{loo} > \epsilon + \beta_m\right] \leq \exp\left(-\frac{2m\epsilon^2}{(m(4\beta_m)+M)^2}\right),$$

which gives (12) by setting the right hand side to δ and using $\delta \leq e^{-1}$. ■

Once again, we notice that the bounds for the empirical error and for the leave-one-out error are very similar. As we will see in later sections, this clearly indicates that our method is not at all suited to the analysis of algorithms which simply perform the minimization of the empirical error (which are not stable in the sense defined above).

4.2.2 Classification Case

In this section we consider the case where $\mathcal{Y} = \{-1, 1\}$ and the algorithm A returns a function A_S that maps instances in $\mathcal{X}$ to labels in $\{-1, 1\}$. The cost function is then simply

$$c(A_S(x), y) = \mathbf{1}_{\{yA_S(x)\leq 0\}}.$$

Thus we see that because of the discrete nature of the cost function, the Uniform Stability of an algorithm with respect to such a cost function can only be $\beta = 0$ or $\beta = 1$. In the first case, it means that the algorithm is always returning the same function. In the second case there is no hope of obtaining interesting bounds since we saw that we need $\beta = O(\frac{1}{m})$ for our bounds to give interesting results.

We thus have to proceed in a different way. One possible approach is to modify our error estimates so that they become "smoother" and have higher stability. The idea to smooth error estimators to decrease their variance is not new and it has even been used in conjunction with McDiarmid's inequality by Lugosi and Pawlak (1994) in order to derive error bounds for certain algorithms. Lugosi and Pawlak studied algorithms which produce estimates for the distributions $P(X|Y = -1)$ and $P(X|Y = +1)$ and defined analogues of the resubstitution and leave-one-out estimates of the error suited to these algorithms.

Here we will take a related, though slightly different route. Indeed, we will consider algorithm having a real-valued output. However, we do not require this output to correspond to a posterior probability but it should simply have the correct sign. That is, the label predicted by such an algorithm is the sign of its real-valued output. Of course, a good algorithm will produce outputs whose absolute value somehow represents the confidence it has in the prediction.

In order to apply the results obtained so far to this setting, we need to introduce some definitions.

Definition 14 *A* real-valued classification algorithm *A is a learning algorithm that maps training sets S to functions $A_S : \mathcal{X} \to \mathbb{R}$ such that the label predicted on an instance x is the sign of $A_S(x)$.*

This class of algorithm includes for instance the classifiers produced by SVM or by ensemble methods such as boosting.

Notice that the cost function defined above extends to the case where the first argument is a real number and have the desired properties: it is zero when the algorithm does predict the right label and 1 otherwise.

Definition 15 (Classification Stability) *A real-valued classification algorithm A has* classification stability *β if the following holds*

$$\forall S \in \mathcal{Z}^m,\ \forall i \in \{1, \ldots, m\},\ \|A_S(.) - A_{S^{\setminus i}}(.)\|_\infty \leq \beta. \tag{14}$$

We introduce a modified cost function:

$$c_\gamma(y,y') = \begin{cases} 1 & \text{for } yy' \leq 0 \\ 1 - yy'/\gamma & \text{for } 0 \leq yy' \leq \gamma \\ 0 & \text{for } yy' \geq \gamma \end{cases}$$

and we denote

$$\ell_\gamma(f,z) = c_\gamma(f(x), y)\,.$$

Accordingly, we define the following error estimates

$$R^\gamma_{emp}(A,S) = \frac{1}{m}\sum_{i=1}^m \ell_\gamma(A_S, z_i)\,,$$

and similarly,

$$R^\gamma_{loo}(A,S) = \frac{1}{m}\sum_{i=1}^m \ell_\gamma(A_{S^{\setminus i}}, z_i)\,.$$

The loss ℓ_γ will count an error each time the function f gives an output close to zero, the closeness being controlled by γ.

Lemma 16 *A real-valued classification algorithm A with classification stability β has uniform stability β/γ with respect to the loss function ℓ_γ.*

Proof It is easy to see that c_γ is $1/\gamma$-Lipschitz with respect to its first argument and so does ℓ_γ by definition. Thus we have for all i, all training set S, and all z,

$$|l_\gamma(A_S, z) - l_\gamma(A_{S^{\setminus i}}, z)| = |c_\gamma(A_S(x), y) - c_\gamma(A_{S^{\setminus i}}(x), y)| \leq \frac{1}{\gamma}|A_S(x) - A_{S^{\setminus i}}(x)| \leq \beta/\gamma\,.$$

■

We can thus apply Theorem 12 with the loss function ℓ_γ and get the following theorem.

Theorem 17 *Let A be a real-valued classification algorithm with stability β. Then, for all $\gamma > 0$, any $m \geq 1$, and any $\delta \in (0,1)$, with probability at least $1-\delta$ over the random draw of the sample S,*

$$R \leq R^\gamma_{emp} + 2\frac{\beta}{\gamma} + \left(4m\frac{\beta}{\gamma} + 1\right)\sqrt{\frac{\ln 1/\delta}{2m}}\,, \tag{15}$$

and with probability at least $1-\delta$ over the random draw of the sample S,

$$R \leq R^\gamma_{loo} + \frac{\beta}{\gamma} + \left(4m\frac{\beta}{\gamma} + 1\right)\sqrt{\frac{\ln 1/\delta}{2m}}\,. \tag{16}$$

Proof We apply Theorem 12 to A with the loss function ℓ_γ which is bounded by $M = 1$ and for which the algorithm is β/γ-stable. Moreover, we use the fact that $R(A_S) \leq R^\gamma = \mathbb{E}_z\left[l_\gamma(A_S, z)\right]$. ■

In order to make this result more practically useful, we need a statement that would hold uniformly for all values γ. The same techniques as in Bartlett (1996) lead to the following result:

Theorem 18 *Let A be a real-valued classification algorithm with stability β and B be some real number. Then, for any $m \geq 1$, and any $\delta \in (0,1)$, with probability at least $1-\delta$ over the random draw of the sample S,*

$$\forall \gamma \in (0, B],\; R \leq R^\gamma_{emp} + 2\frac{e\beta}{\gamma} + \left(\frac{4me\beta}{\gamma} + 1\right)\sqrt{\frac{1}{2m}}\left(\sqrt{\ln 1/\delta} + \sqrt{2\ln\ln\frac{eB}{\gamma}}\right)\,, \tag{17}$$

and

$$\forall \gamma \in (0, B],\ R \leq R_{loo}^{\gamma} + \frac{e\beta}{\gamma} + \left(\frac{4me\beta}{\gamma} + 1\right)\sqrt{\frac{1}{2m}}\left(\sqrt{\ln 1/\delta} + \sqrt{2 \ln \ln \frac{eB}{\gamma}}\right), \tag{18}$$

We defer the proof of this theorem to the appendix.

We can thus apply Theorem 18 with a value of γ which is optimized after having seen the data.

5. Stable Learning Algorithms

As seen in previous sections, our approach allowed to derive bounds on the generalization error from the empirical and leave-one-out errors which depend on the stability of the algorithm. However, we noticed that the bounds we obtain for the two estimators are very similar. This readily implies that the method is suited to the study of algorithms for which the empirical error is close to the leave-one-out error. There is thus no hope to get good bounds for algorithms which simply minimize the empirical error since their empirical error will be very much optimistically biased compared to their leave-one-out error.

This means that, in order to be stable in the sense defined above, a learning algorithm has to significantly depart from an empirical risk minimizer. It thus has to accept a significant number of training errors (which should however not be larger that the noise level). In order to generalize, these extra training errors will thus be compensated by a decrease of the complexity of the learned function.

In some sense, this is exactly what regularization-based algorithm do: they minimize an objective function which is the sum of an empirical error term and a regularizing term which penalizes the complexity of the solution. This explains why our approach is particularly well suited for the analysis of such algorithms.

5.1 Previous Results for k-Local Rules

As an illustration of the various notions of stability, we will first study the case of k-Local Rules for which a large number of results were obtained.

A k-Local Rule is a classification algorithm that determines the label of an instance x based on the k closest instances in the training set. The simplest example of such a rule is the k-Nearest Neighbors (k-NN) algorithm which computes the label by a majority vote among the labels of the k nearest instances in the training set. Such an algorithm can be studied as a $\{0,1\}$-valued classifier or as a $[0,1]$-valued classifier if we take into account the result of the vote.

We will consider the real-valued version of the k-NN classifier and give a result about its stability with respect to different loss functions.

1. With respect to the $\{0,1\}$-loss function, the k-NN classifier has hypothesis stability

   $$\beta \leq \frac{4}{m}\sqrt{\frac{k}{2\pi}}.$$

   This was proven in Devroye and Wagner (1979a). We will not reproduce the proof which is quite technical but notice that a symmetry argument readily gives

   $$\mathbb{P}\left[A_S(z) \neq A_{S^{\setminus i}}(z)\right] \leq \frac{k}{m}.$$

2. With respect to the absolute loss function ($c(y, y') = |y - y'|$), the k-NN classifier has only a trivial uniform stability which is the bound on the values of y.

The polynomial bound that can be obtained from hypothesis stability suggests that k should be small if one wants a good bound. This is somehow counter-intuitive since the decision seems more robust to noise when many points are involved in the vote. There exist exponential bounds on the leave-one-out estimate of k-NN for the $\{0,1\}$-loss obtained by Devroye and Wagner (1979a) and for the smoothed error estimate (i.e. with respect to the absolute loss) obtained by Lugosi and Pawlak (1994), and these bounds do not depend on the parameter k (due to a more careful application of McDiarmid's inequality suited to the algorithm). We may then wonder in that case whether the polynomial bounds are interesting compared to exponential ones since the latter are sharper and are closer to intuitive interpretation. Despite this example, we believe that in general polynomial bounds could give relevant hints about which feature of the learning algorithm leads to good generalization.

In the remainder, we will consider several algorithms that have not been studied from a stability perspective and we will focus on their uniform stability only, which turns out to be quite good. Obtaining results directly for their hypothesis stability remains an open problem.

5.2 Stability of Regularization Algorithms

Uniform stability may appear as a strict condition. Actually, we will see in this section that many existing learning methods exhibit a uniform stability which is controlled by the regularization parameter and can thus be very small.

5.2.1 Stability for General Regularizers

Recall that $\ell(f,z) = c(f(x),y)$. We assume in this section that $\mathcal{F}$ is a convex subset of a linear space.

Definition 19 *A loss function ℓ defined on $\mathcal{F}\times\mathcal{Y}$ is σ-admissible with respect to $\mathcal{F}$ if the associated cost function c is convex with respect to its first argument and the following condition holds*

$$\forall y_1, y_2 \in \mathcal{D},\ \forall y' \in \mathcal{Y}, |c(y_1,y') - c(y_2,y')| \leq \sigma |y_1 - y_2|\,,$$

where $\mathcal{D} = \{y : \exists f \in \mathcal{F}, \exists x \in \mathcal{X}, f(x) = y\}$ is the domain of the first argument of c.

Thus in the case of the quadratic loss for example, this condition is verified if $\mathcal{Y}$ is bounded and $\mathcal{F}$ is totally bounded, that is there exists $M < \infty$ such that

$$\forall f \in \mathcal{F},\ \|f\|_\infty \leq M \ \text{ and } \ \forall y \in \mathcal{Y},\ |y| \leq M\,.$$

We introduce the objective function that the algorithm will minimize: let $N : \mathcal{F} \to R_+$ be a function on $\mathcal{F}$,

$$R_r(g) := \frac{1}{m}\sum_{j=1}^{m} \ell(g,z_j) + \lambda N(g)\,, \tag{19}$$

and a modified version (based on a truncated training set),

$$R_r^{\setminus i}(g) := \frac{1}{m}\sum_{j\neq i} \ell(g,z_j) + \lambda N(g)\,. \tag{20}$$

Depending on the algorithm N will take different forms. To derive stability bounds, we need some general results about the minimizers of (19) and (20).

Lemma 20 *Let ℓ be σ-admissible with respect to $\mathcal{F}$, and N a functional defined on $\mathcal{F}$ such that for all training sets S, R_r and $R_r^{\setminus i}$ have a minimum (not necessarily unique) in $\mathcal{F}$. Let f denote a* *minimizer in $\mathcal{F}$ of R_r, and for $i = 1, \ldots, m$, let $f^{\setminus i}$ denote a minimizer in $\mathcal{F}$ of $R_r^{\setminus i}$. We have for any $t \in [0, 1]$,*

$$N(f) - N(f + t\Delta f) + N(f^{\setminus i}) - N(f^{\setminus i} - t\Delta f) \leq \frac{t\sigma}{\lambda m} |\Delta f(x_i)| \,, \tag{21}$$

where $\Delta f = f^{\setminus i} - f$.

Proof
Let us introduce the notation

$$R_{emp}^{\setminus i}(f) := \frac{1}{m} \sum_{j \neq i} \ell(f, z_j) \,.$$

Recall that a convex function g verifies:

$$\forall x, y, \, \forall t \in [0, 1] \quad g(x + t(y - x)) - g(x) \leq t(g(y) - g(x)) \,.$$

Since c is convex, $R_{emp}^{\setminus i}$ is convex too and thus, $\forall t \in [0, 1]$

$$R_{emp}^{\setminus i}(f + t\Delta f) - R_{emp}^{\setminus i}(f) \leq t(R_{emp}^{\setminus i}(f^{\setminus i}) - R_{emp}^{\setminus i}(f)) \,.$$

We can also get (switching the role of f and $f^{\setminus i}$):

$$R_{emp}^{\setminus i}(f^{\setminus i} - t\Delta f) - R_{emp}^{\setminus i}(f^{\setminus i}) \leq t(R_{emp}^{\setminus i}(f) - R_{emp}^{\setminus i}(f^{\setminus i})) \,.$$

Summing the two preceding inequalities yields

$$R_{emp}^{\setminus i}(f + t\Delta f) - R_{emp}^{\setminus i}(f) + R_{emp}^{\setminus i}(f^{\setminus i} - t\Delta f) - R_{emp}^{\setminus i}(f^{\setminus i}) \leq 0 \,. \tag{22}$$

Now, by assumption we have

$$R_r(f) - R_r(f + t\Delta f) \leq 0 \tag{23}$$
$$R_r^{\setminus i}(f^{\setminus i}) - R_r^{\setminus i}(f^{\setminus i} - t\Delta f) \leq 0 \,, \tag{24}$$

so that, summing the two previous inequalities and using (22), we get

$$c(f(x_i), y_i) - c((f + t\Delta f)(x_i), y_i) + m\lambda \Big(N(f) - N(f + t\Delta f) + N(f^{\setminus i}) - N(f^{\setminus i} - t\Delta f) \Big) \leq 0 \,,$$

and thus, by the σ-admissibility condition, we get

$$N(f) - N(f + t\Delta f) + N(f^{\setminus i}) - N(f^{\setminus i} - t\Delta f) \leq \frac{t\sigma}{\lambda m} |\Delta f(x_i)| \,.$$

■

In the above lemma, there is no assumption about the space $\mathcal{F}$ (apart from being a convex linear space) and the regularizer N apart from the existence of minima for R_r and $R_r^{\setminus i}$. However, most of the practical regularization-based algorithms work with a space $\mathcal{F}$ that is a vector space and with a convex regularizer. We will thus refine our previous result in this particular setting. In order to do this, we need some standard definitions about convex functions which we deferred to Appendix C where most of the material can be found in Rockafellar (1970) and in Gordon (1999).

Lemma 21 *Under the conditions of Lemma 20, when $\mathcal{F}$ is a vector space and N is a proper closed convex function from $\mathcal{F}$ to $R \cup \{-\infty, +\infty\}$, we have*

$$d_N(f, f^{\setminus i}) + d_N(f^{\setminus i}, f) \leq \frac{1}{\lambda m} \Big(\ell(f^{\setminus i}, z_i) - \ell(f, z_i) - d_{\ell(.,z_i)}(f^{\setminus i}, f) \Big) \leq \frac{\sigma}{\lambda m} |\Delta f(x_i)| \,,$$

when N and ℓ are differentiable.

Proof We start with the differentiable case and work with regular divergences. By definition of f and $f^{\setminus i}$, we have, using (30),

$$d_{R_r}(f^{\setminus i}, f) + d_{R_r^{\setminus i}}(f, f^{\setminus i}) = R_r(f^{\setminus i}) - R_r(f) + R_r^{\setminus i}(f) - R_r^{\setminus i}(f^{\setminus i}) = \frac{1}{m}\ell(f^{\setminus i}, z_i) - \frac{1}{m}\ell(f, z_i)\,.$$

Moreover, by the nonnegativity of divergences, we have

$$d_{R_{emp}^{\setminus i}}(f, f^{\setminus i}) + d_{R_{emp}^{\setminus i}}(f^{\setminus i}, f) \geq 0\,,$$

which, with the previous equality and the fact that $d_{A+B} = d_A + d_B$, gives

$$\lambda d_N(f, f^{\setminus i}) + \lambda d_N(f^{\setminus i}, f) \leq \frac{1}{m}\left(\ell(f^{\setminus i}, z_i) - \ell(f, z_i) - d_{\ell(.,z_i)}(f^{\setminus i}, f)\right)\,,$$

and we obtain the first part of the result. For the second part, we notice that

$$\ell(f^{\setminus i}, z_i) - \ell(f, z_i) - d_{\ell(.,z_i)}(f^{\setminus i}, f) \leq \ell(f^{\setminus i}, z_i) - \ell(f, z_i)\,,$$

by the nonnegativity of the divergence and thus

$$\ell(f^{\setminus i}, z_i) - \ell(f, z_i) - d_{\ell(.,z_i)}(f^{\setminus i}, f) \leq \sigma|f^{\setminus i}(x_i) - f(x_i)|\,,$$

by the σ-admissibility condition. ■

The results in this section can be used to derive bounds on the stability of many learning algorithms. Each procedure that can be interpreted as the minimization of a regularized functional can be analyzed with these lemmas. The only thing that will change from one procedure to another is the regularizer N and the cost function c. In the following, we show how to apply these theorems to different learning algorithms.

5.2.2 Application to Regularization in Hilbert Spaces

Many algorithms such as Support Vector Machines (SVM) or classical regularization networks introduced by Poggio and Girosi (1990) perform the minimization of a regularized objective function where the regularizer is a norm in a reproducing kernel Hilbert space (RKHS):

$$N(f) = \|f\|_k^2\,,$$

where k refers to the kernel (see e.g. Wahba, 2000, or Evgeniou et al., 1999, for definitions). The fundamental property of a RKHS $\mathcal{F}$ is the so-called reproducing property which writes

$$\forall f \in \mathcal{F},\ \forall x \in \mathcal{X},\ f(x) = \langle f, k(x,.)\rangle\ .$$

In particular this gives by Cauchy-Schwarz inequality

$$\forall f \in \mathcal{F},\ \forall x \in \mathcal{X},\ |f(x)| \leq \|f\|_k\sqrt{k(x,x)}\,. \tag{25}$$

We now state a result about the uniform stability of RKHS learning.

Theorem 22 *Let $\mathcal{F}$ be a reproducing kernel Hilbert space with kernel k such that $\forall x \in \mathcal{X},\ k(x,x) \leq \kappa^2 < \infty$. Let ℓ be σ-admissible with respect to $\mathcal{F}$. The learning algorithm A defined by*

$$A_S = \arg\min_{g\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \ell(g, z_i) + \lambda\|g\|_k^2\,, \tag{26}$$

has uniform stability β with respect to ℓ with

$$\beta \leq \frac{\sigma^2\kappa^2}{2\lambda m}\,.$$

Proof We use the proof technique described in previous section. It can be easily checked that when $N(.) = \|.\|_k^2$ we have

$$d_N(g, g') = \|g - g'\|_k^2 \,.$$

Thus, Lemma 20 gives

$$2\|\Delta f\|_k^2 \leq \frac{\sigma}{\lambda m} |\Delta f(x_i)| \,.$$

Using (25), we get

$$|\Delta f(x_i)| \leq \|\Delta f\|_k \sqrt{k(x_i, x_i)} \leq \kappa \|\Delta f\|_k \,,$$

so that

$$\|\Delta f\|_k \leq \frac{\kappa\sigma}{2\lambda m} \,.$$

Now we have, by the σ-admissibility of ℓ

$$|\ell(f, z) - \ell(f^{\setminus i}, z)| \leq \sigma |f(x) - f^{\setminus i}(x)| = \sigma |\Delta f(x)| \,,$$

which, using (25) again, gives the result. ■

We are now one step away from being able to apply Theorem 12. The only thing that we need is to bound the loss function. Indeed, the σ-admissibility condition does not ensure the boundedness. However, since we are in a RKHS, we can use the following simple lemma which ensures that if we have an a priori bound on the target values y, then the boundedness condition is satisfied.

Lemma 23 *Let A be the algorithm of Theorem 22 where ℓ is a loss function associated to a convex cost function $c(.,.)$. We denote by $B(.)$ a positive non-decreasing real-valued function such that for all $y \in \mathcal{D}$.*

$$\forall y' \in \mathcal{Y},\; c(y, y') \leq B(y)$$

For any training set S, we have

$$\|f\|_k^2 \leq \frac{B(0)}{\lambda} \,,$$

and also

$$\forall z \in \mathcal{Z},\; 0 \leq \ell(A_S, z) \leq B\left(\kappa \sqrt{\frac{B(0)}{\lambda}}\right) \,.$$

Moreover, ℓ is σ-admissible where σ can be taken as

$$\sigma = \sup_{y' \in \mathcal{Y}} \sup_{|y| \leq B\left(\kappa\sqrt{\frac{B(0)}{\lambda}}\right)} \left| \frac{\partial c}{\partial y}(y, y') \right| \,.$$

Proof We have for $f = A_S$,

$$R_r(f) \leq R_r(\vec{0}) = \frac{1}{m} \sum_{i=1}^m \ell(\vec{0}, z_i) \leq B(0) \,,$$

and also $R_r(f) \geq \lambda \|f\|_k^2$ which gives the first inequality. The second inequality follows from (25). The last one is a consequence of the definition of σ-admissibility. ■

Example 1 (Stability of bounded SVM regression) *Assume k is a bounded kernel, that is $k(x, x) \leq \kappa^2$ and $\mathcal{Y} = [0, B]$. Consider the loss function*

$$\ell(f, z) = |f(x) - y|_\epsilon = \begin{cases} 0 & \text{if } |f(x) - y| \leq \epsilon \\ |f(x) - y| - \epsilon & \text{otherwise} \end{cases}$$

This function is 1*-admissible and we can state* $B(y) = B$. *The SVM algorithm for regression with a kernel* k *can be defined as*

$$A_S = \arg\min_{g \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \ell(g, z_i) + \lambda \|g\|_k^2 ,$$

and we thus get the following stability bound

$$\beta \leq \frac{\kappa^2}{2\lambda m} .$$

Moreover, by Lemma 23 we have

$$\forall z \in \mathcal{Z},\, 0 \leq \ell(A_S, z) \leq \kappa \sqrt{\frac{B}{\lambda}}$$

Plugging the above into Theorem 12 gives the following bound

$$R \leq R_{emp} + \frac{\kappa^2}{\lambda m} + \left(\frac{2\kappa^2}{\lambda} + \kappa \sqrt{\frac{B}{\lambda}} \right) \sqrt{\frac{\ln 1/\delta}{2m}} .$$

Note that we consider here SVM without the bias b*, which is strictly speaking different from the true definition of SVM. The question whether* b *can be included in such a setting remains open.*

Example 2 (Stability of soft margin SVM classification) *We have* $\mathcal{Y} = \{-1, 1\}$. *We consider the following loss function*

$$\ell(f, z) = (1 - y f(x))_+ = \begin{cases} 1 - y f(x) & \text{if } 1 - y f(x) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

which is 1*-admissible. From Lemma 20, we deduce that the real-valued classification obtained by the SVM optimization procedure has classification stability* β *with*

$$\beta \leq \frac{\kappa^2}{2\lambda m} .$$

We use Theorem 17 with $\gamma = 1$ *and thus get*

$$R \leq R_{emp}^1 + \frac{\kappa^2}{\lambda m} + \left(1 + \frac{2\kappa^2}{\lambda} \right) \sqrt{\frac{\ln 1/\delta}{2m}} ,$$

where R_{emp}^1 *is the clipped error. It can be seen that* $R_{emp}^1 \leq \frac{1}{m} \sum_{i=1}^{m} \ell(f, z_i) = \frac{1}{m} \sum_{i=1}^{m} \xi_i$*, where the* ξ *are the Lagrange multipliers that appear in the dual formulation of the soft-margin SVM.*

Note that the same remark as in the previous example holds here: there is no bias b *in the definition of the SVM.*

Example 3 (Stability of Regularized Least Squares Regression) *Again we will consider the bounded case* $\mathcal{Y} = [0, B]$. *The regularized least squares regression algorithm is defined by*

$$A_S = \arg\min_{g \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \ell(g, z_i) + \lambda \|g\|_k^2 ,$$

where $\ell(f, z) = (f(x) - y)^2$. *We can state* $B(y) = B^2$ *so that* ℓ *is* $2B$*-admissible by Lemma 23. Also we have*

$$\forall z \in \mathcal{Z},\, 0 \leq \ell(A_S, z) \leq \kappa \sqrt{\frac{B}{\lambda}} .$$

The stability bound for this algorithm is thus

$$\beta \leq \frac{2\kappa^2 B^2}{\lambda m}$$

so that we have the generalization error bound

$$R \leq R_{emp} + \frac{4\kappa^2 B^2}{\lambda m} + \left(\frac{8\kappa^2 B^2}{\lambda} + 2B\right)\sqrt{\frac{\ln 1/\delta}{2m}}\,.$$

5.2.3 Regularization by the Relative Entropy

In this section we consider algorithms that build a mixture or a weighted combination of base hypotheses.

Let's consider a set $\mathcal{H}$ of functions $h : \mathcal{X} \to \mathcal{Y}$ parameterized by some parameter θ:

$$\mathcal{H} = \{h_\theta : \theta \in \Theta\}\,.$$

This set is the base class from which the learning algorithm will form mixtures by averaging the predictions of base hypotheses. More precisely, we assume that Θ is a measurable space where a reference measure is defined. The output of our algorithm is a mixture of element from Θ, in other words, it is a probability distribution over Θ. We will thus choose $\mathcal{F}$ as the set of all such probability distributions (dominated by the reference measure), defined by their density with respect to the reference measure.

Once an element $f \in \mathcal{F}$ is chosen by the algorithm, the predictions are computed as follows

$$\hat{y}(x) = \int_\Theta h_\theta(x) f(\theta) d\theta\,,$$

which means that the prediction produced by the algorithm is indeed a weighted combination of the predictions of the base hypotheses, weighted by the density f. In Bayesian terms, A_S would be a posterior on Θ computed from the observation of S and $\hat{y}(x)$ is the corresponding Bayes prediction.

By some abuse of notation, we will denote by A_S both the element $f \in \mathcal{F}$ that is used by the algorithm to weigh the base hypotheses (which can be considered as a function $\Theta \to R$) and the prediction function $x \in \mathcal{X} \mapsto \hat{y}(x)$.

Now we need to define a loss function on $\mathcal{F} \times \mathcal{Z}$. This can be done by extending a loss function r defined on $\mathcal{H} \times \mathcal{Z}$ with associated cost function s ($r(h,z) = s(h(x), y)$). There are two ways of deriving a loss function on $\mathcal{F}$. We can simply use s to compute the discrepancy between the predicted and true labels

$$\ell(g, z) = s(\hat{y}(x), y)\,, \tag{27}$$

or we can average the loss over Θ,

$$\ell(g, z) = \int_\Theta r(h_\theta, z) g(\theta) d\theta\,. \tag{28}$$

The first loss is the one used when one is doing Bayesian averaging of hypotheses. The second loss corresponds to the expected loss of a randomized algorithm that would sample $h \in \mathcal{H}$ according to the posterior A_S to perform the predictions.

In the remainder, we will focus on the second type of loss since it is easier to analyze. Note however, that this loss will be used only to define a regularization algorithm and that the loss that is used to measure its error may be different.

Our goal is to choose the posterior f via the minimization of a regularized objective function. We choose some fixed density f_0 and define the regularizer as

$$N(g) = K(g, f_0) = \int_\Theta g(\theta) \ln \frac{g(\theta)}{f_0(\theta)} d\theta\,,$$

K being the Kullback-Leibler divergence or the relative entropy. In Bayesian terms, f_0 would be our *prior*. Now, the goal is to minimize the following objective function

$$R_r(g) = \frac{1}{m}\sum_{i=1}^{m} \ell(g,z) + \lambda K(g,f_0)\,,$$

where ℓ is given by (28). We can interpret the minimization of this objective function as the computation of the Maximum A Posteriori (MAP) estimate.

Let's analyze this algorithm. We will assume that we know a bound M on the loss $r(h_\theta, z)$. First, notice that ℓ is linear in g and is thus convex and M-Lipschitz with respect to the L_1 norm

$$|\ell(g,z) - \ell(g',z)| \le M \int_\Theta |g(\theta) - g'(\theta)| d\theta\,.$$

Thus ℓ is M-admissible with respect to $\mathcal{F}$.

We can now state the following result on the uniform stability of the algorithm defined above.

Theorem 24 *Let $\mathcal{F}$ defined as above and let r be any loss function defined on $\mathcal{H} \times \mathcal{Z}$, bounded by M. Let f_0 be a fixed member of $\mathcal{F}$. When ℓ is defined by (28), the learning algorithm A defined by*

$$A_S = \arg\min_{g\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \ell(g,z_i) + \lambda K(g,f_0)\,, \tag{29}$$

has uniform stability β with respect to ℓ with

$$\beta \le \frac{M^2}{\lambda m}\,.$$

Proof Recall the following property of the relative entropy (see e.g. Cover and Thomas 1991), for any g, g',

$$\frac{1}{2}\left(\int_\Theta |g(\theta) - g'(\theta)| d\theta\right)^2 \le K(g,g')\,.$$

Moreover, the Bregman divergence associated to the relative entropy to f_0 is

$$d_{K(.,f_0)}(g,g') = K(g,g')\,.$$

We saw that ℓ is M-admissible thus, by Lemma 21 we get

$$\left(\int_\Theta |f(\theta) - f^{\setminus i}(\theta)| d\theta\right)^2 \le \frac{M}{\lambda m}\int_\Theta |f(\theta) - f^{\setminus i}(\theta)| d\theta\,,$$

hence

$$\int_\Theta |f(\theta) - f^{\setminus i}(\theta)| d\theta \le \frac{M}{\lambda m}\,,$$

and thus, using again the M-admissibility of ℓ, we get for all $z \in \mathcal{Z}$,

$$|\ell(f,z) - \ell(f^{\setminus i},z)| \le \frac{M^2}{\lambda m}\,,$$

which concludes the proof. ■

Now, let's consider the case of classification where $\mathcal{Y} = \{-1, 1\}$. If we use base hypotheses h_θ that return values in $\{-1,1\}$, it is easy to see from the proof of the above theorem that algorithm A has classification stability $\beta \le \frac{M}{\lambda m}$. Indeed, we have

$$|A_S(x) - A_{S^{\setminus i}}(x)| = \left|\int_\Theta h_\theta(x)(A_S(\theta) - A_{S^{\setminus i}}(\theta)) d\theta\right| \le \int_\Theta |A_S(\theta) - A_{S^{\setminus i}}(\theta)| d\theta \le \frac{M}{\lambda m}\,,$$

where the last inequality is derived in the proof of Theorem 24.

Example 4 (Maximum Entropy Discrimination) *Jaakola et al. (1999) introduce the Minimum Relative Entropy (MRE) algorithm which is a real-valued classifier obtained by minimizing*

$$R_r(g) = \frac{1}{m}\sum_{i=1}^{m} \ell(g, z) + \lambda K(g, f_0)\,,$$

where the base class has two parameters $\mathcal{H} = \{h_{\theta,\gamma} : \theta \in \Theta, \gamma \in R\}$ (with $h_{\theta,\gamma} = h_\theta$) and the loss is defined by

$$\ell(g, z) = \left(\int_{\Theta,R} (\gamma - y h_\theta(x)) g(\theta) d\theta d\gamma\right)_+ .$$

If we have a bound B on the quantity $\gamma - y h_\theta(x)$, we see that this loss function is B-admissible and thus by Theorem 24 (and the remark about the classification stability) we deduce that the MRE algorithm has classification stability β bounded by

$$\beta \leq \frac{B}{\lambda m}$$

6. Discussion

For regularization algorithms, we obtained bounds on the uniform stability of the order of $\beta = O(\frac{1}{\lambda m})$. Plugging this result into our main theorem, we obtained bounds on the generalization error of the following type

$$R \leq R_{emp} + O\left(\frac{1}{\lambda\sqrt{m}}\right)\,,$$

so that we obtain non trivial results only if we can guarantee that $\lambda >> \frac{1}{\sqrt{m}}$. This is likely to depend on the noise in the data and no theoretical results exist that guarantee that λ does not decrease too fast when m is increased.

However, it should be possible to refine our results which used sometimes quite crude bounds. It seems reasonable that a bound like

$$R \leq R_{emp} + O\left(\frac{1}{\sqrt{\lambda m}}\right)\,,$$

could be possible to obtain. This remains an open problem.

In order to better understand the distinctive feature of our bounds, we can compare them to bounds from Structural Risk Minimization (SRM) for example on the SVM algorithm. The SVM algorithm can be presented using the two equivalent formulations

$$\min_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}(1 - y_i f(x_i))_+ + \lambda\|f\|^2\,,$$

or

$$\min_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m}(1 - y_i f(x_i))_+ \text{ with } \|f\|^2 \leq \nu\,,$$

The equivalence of those two problems comes from the fact that for any λ, there exists a ν such that the solution of the two problems are the same.

The SRM principle consists in solving the second problem for several values of ν and then choosing the value that minimizes a bound that depends on the VC-dimension of the set $\{f : \|f\|^2 \leq \nu\}$. However, this quantity is usually not easy to compute and only loose upper bounds can be found. Moreover, since minimization under a constraint on the norm is not easy to perform, one typically performs the first minimization for a particular value of λ (chosen by cross-validation) and then uses SRM bounds with $\nu = \|f\|^2$. This requires the SRM bounds to hold uniformly for all values of ν.

This approach has led to bound which were quite predictive of the behavior but that were quantitatively very loose.

In contrast, our approach directly focuses on the actual minimization that is performed (the first one) and does not require the computation of a complexity measure. Indeed, the complexity is implicitly evaluated by the actual parameter λ.

7. Conclusion

We explored the possibility of obtaining generalization bounds for specific algorithms from stability properties. We introduced several notions of stability and obtained corresponding generalization bounds with either the empirical error or the leave-one-out error. Our main result is an exponential bound for algorithms that have good uniform stability. We then proved that regularization algorithms have such a property and that their stability is controlled by the regularization parameter λ. This allowed us to obtained bounds on the generalization error of Support Vector Machines both in the classification and in the regression framework that do not depend on the implicit VC-dimension but rather depend explicitly on the tradeoff parameter C.

Further directions of research include the question of obtaining better bounds via uniform stability and the use of less restrictive notions of stability. Of great practical interest would be to design algorithms that maximize their own stability.

Acknowledgements

The authors wish to thank Ryan Rifkin and Ralf Herbrich for fruitful comments that helped improved the readability and Alex Smola, Gábor Lugosi, Stéphane Boucheron and Sayan Mukherjee for stimulating discussions.

Appendix A. Proof of Lemma 9

Let's start with a generalized version of a lemma from Rogers and Wagner (1978).

Lemma 25 *For any learning algorithm A, any $i, j \in \{1, \ldots, m\}$ such that $i \neq j$, we have*

$$\begin{aligned}
\mathbb{E}_S\left[(R - R_{emp})^2\right] \leq\ & \mathbb{E}_{S,z,z'}\left[\ell(A_S, z)\ell(A_S, z')\right] - 2\mathbb{E}_{S,z}\left[\ell(A_S, z)\ell(A_S, z_i)\right] \\
& + \mathbb{E}_S\left[\ell(A_{S^{\setminus i}}, z_i)\ell(A_{S^{\setminus j}}, z_j)\right] + \frac{M}{m}\mathbb{E}_S\left[\ell(A_S, z_i)\right] \\
& - \frac{1}{m}\mathbb{E}_S\left[\ell(A_S, z_i)\ell(A_S, z_j)\right] ,
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}_S\left[(R - R_{loo})^2\right] \leq\ & \mathbb{E}_{S,z,z'}\left[\ell(A_S, z)\ell(A_S, z')\right] - 2\mathbb{E}_{S,z}\left[\ell(A_S, z)\ell(A_{S^{\setminus i}}, z_i)\right] \\
& + \mathbb{E}_S\left[\ell(A_{S^{\setminus i}}, z_i)\ell(A_{S^{\setminus j}}, z_j)\right] + \frac{M}{m}\mathbb{E}_S\left[R^{\setminus i}\right] \\
& - \frac{1}{m}\mathbb{E}_S\left[\ell(A_{S^{\setminus i}}, z_i)\ell(A_{S^{\setminus j}}, z_j)\right] ,
\end{aligned}$$

Proof We have

$$\begin{aligned}
\mathbb{E}_S\left[R^2\right] &= \mathbb{E}_S\left[\mathbb{E}_z\left[\ell(A_S, z)\right]^2\right] \\
&= \mathbb{E}_S\left[\mathbb{E}_z\left[\ell(A_S, z)\right]\mathbb{E}_{z'}\left[\ell(A_S, z')\right]\right] \\
&= \mathbb{E}_S\left[\mathbb{E}_{z,z'}\left[\ell(A_S, z)\ell(A_S, z')\right]\right] ,
\end{aligned}$$

and also

$$\begin{aligned}
\mathbb{E}_S\left[RR_{emp}\right] &= \mathbb{E}_S\left[R\frac{1}{m}\sum_{i=1}^m \ell(A_S,z_i)\right] \\
&= \frac{1}{m}\sum_{i=1}^m \mathbb{E}_S\left[R\ell(A_S,z_i)\right] \\
&= \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{S,z}\left[\ell(A_S,z)\ell(A_S,z_i)\right] \\
&= \mathbb{E}_{S,z}\left[\ell(A_S,z)\ell(A_S,z_i)\right] ,
\end{aligned}$$

and also

$$\begin{aligned}
\mathbb{E}_S\left[RR_{loo}\right] &= \mathbb{E}_S\left[R\frac{1}{m}\sum_{i=1}^m \ell(A_{S^{\setminus i}},z_i)\right] \\
&= \frac{1}{m}\sum_{i=1}^m \mathbb{E}_S\left[R\ell(A_{S^{\setminus i}},z_i)\right] \\
&= \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{S,z}\left[\ell(A_S,z)\ell(A_{S^{\setminus i}},z_i)\right] \\
&= \mathbb{E}_{S,z}\left[\ell(A_S,z)\ell(A_{S^{\setminus i}},z_i)\right] ,
\end{aligned}$$

for any fixed i by symmetry. Also we have

$$\begin{aligned}
\mathbb{E}_S\left[R_{emp}^2\right] &= \frac{1}{m^2}\sum_{i=1}^m \mathbb{E}_S\left[\ell(A_S,z_i)^2\right] + \frac{1}{m^2}\sum_{i\neq j}\mathbb{E}_S\left[\ell(A_S,z_i)\ell(A_S,z_j)\right] \\
&\leq \frac{M}{m}\mathbb{E}_S\left[\frac{1}{m}\sum_{i=1}^m \ell(A_S,z_i)\right] + \frac{m-1}{m}\mathbb{E}_S\left[\ell(A_S,z_i)\ell(A_S,z_j)\right] \\
&= \frac{M}{m}\mathbb{E}_S\left[\ell(A_S,z_i)\right] + \frac{m-1}{m}\mathbb{E}_S\left[\ell(A_S,z_i)\ell(A_S,z_j)\right] ,
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}_S\left[R_{loo}^2\right] &= \frac{1}{m^2}\sum_{i=1}^m \mathbb{E}_S\left[\ell(A_{S^{\setminus i}},z_i)^2\right] + \frac{1}{m^2}\sum_{i\neq j}\mathbb{E}_S\left[\ell(A_{S^{\setminus i}},z_i)\ell(A_{S^{\setminus j}},z_j)\right] \\
&\leq \frac{M}{m}\mathbb{E}_S\left[\frac{1}{m}\sum_{i=1}^m \ell(A_{S^{\setminus i}},z_i)\right] + \frac{m-1}{m}\mathbb{E}_S\left[\ell(A_{S^{\setminus i}},z_i)\ell(A_{S^{\setminus j}},z_j)\right] \\
&= \frac{M}{m}\mathbb{E}_S\left[R^{\setminus i}\right] + \frac{m-1}{m}\mathbb{E}_S\left[\ell(A_{S^{\setminus i}},z_i)\ell(A_{S^{\setminus j}},z_j)\right] .
\end{aligned}$$

which concludes the proof. ∎

Now let's prove Lemma 9. We will use several times the fact that the random variables are i.i.d. and we can thus interchange them without modifying the expectation (it is just a matter of renaming them). We introduce the notation $T = S^{\setminus i,j}$ and we will denote by $A_{T,z,z'}$ the result of training on the set $T \cup z, z'$.

Let's first formulate the first inequality of Lemma (25) as

$$\mathbb{E}_S\left[(R-R_{emp})^2\right] \leq \frac{1}{m}\mathbb{E}_S\left[\ell(A_S,z_i)\left(M-\ell(A_S,z_j)\right)\right]$$

$$
\begin{aligned}
&+\mathbb{E}_{S,z,z'}\left[\ell(A_S,z)\ell(A_S,z') - \ell(A_S,z)\ell(A_S,z_i)\right]\\
&+\mathbb{E}_{S,z,z'}\left[\ell(A_S,z_i)\ell(A_S,z_j) - \ell(A_S,z)\ell(A_S,z_i)\right]\\
&= I_1 + I_2 + I_3\,.
\end{aligned}
$$

Using Schwarz's inequality we have

$$
\begin{aligned}
\mathbb{E}_S\left[\ell(A_S,z_i)\left(M-\ell(A_S,z_j)\right)\right]^2 &\leq \mathbb{E}_S\left[\ell(A_S,z_i)^2\right]\mathbb{E}_S\left[(M-\ell(A_S,z_j))^2\right]\\
&\leq M^2\mathbb{E}_S\left[\ell(A_S,z_i)\right]\mathbb{E}_S\left[M-\ell(A_S,z_j)\right]\\
&= M^2\mathbb{E}_S\left[\ell(A_S,z_i)\right]\left(M-\mathbb{E}_S\left[\ell(A_S,z_i)\right]\right)\\
&\leq \frac{M^4}{4}\,,
\end{aligned}
$$

so that we conclude

$$
I_1 \leq \frac{M^2}{2m}\,.
$$

Now we rewrite I_2 as

$$
\begin{aligned}
&\mathbb{E}_{S,z,z'}\left[\ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z_i)\right]\\
&\quad= \mathbb{E}_{S,z,z'}\left[\ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z_j,z'},z)\ell(A_{T,z_j,z'},z')\right]\\
&\quad\text{(renaming } z_i \text{ as } z' \text{ in the second term)}\\
&\quad= \mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z_j},z) - \ell(A_{T,z,z_j},z))\ell(A_{T,z_i,z_j},z')\right]\\
&\quad+\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z,z_j},z) - \ell(A_{T,z_j,z'},z))\ell(A_{T,z_i,z_j},z')\right]\\
&\quad+\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z_j,z'},z'))\ell(A_{T,z_j,z'},z)\right]\,.
\end{aligned}
$$

Next we rewrite I_3 as

$$
\begin{aligned}
&\mathbb{E}_{S,z,z'}\left[\ell(A_{T,z_i,z_j},z_i)\ell(A_{T,z_i,z_j},z_j) - \ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z_i)\right]\\
&\quad= \mathbb{E}_{S,z,z'}\left[\ell(A_{T,z,z'},z)\ell(A_{T,z,z'},z') - \ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z_i)\right]\\
&\quad\text{(renaming } z_j \text{ as } z' \text{ and } z_i \text{ as } z \text{ in the first term)}\\
&\quad= \mathbb{E}_{S,z,z'}\left[\ell(A_{T,z,z'},z)\ell(A_{T,z,z'},z') - \ell(A_{T,z',z_i},z)\ell(A_{T,z',z_i},z')\right]\\
&\quad\text{(exchanging } z_i \text{ and } z_j\text{, then renaming } z_j \text{ as } z' \text{ in the second term)}\\
&\quad= \mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z,z'},z') - \ell(A_{T,z,z_i},z'))\ell(A_{T,z,z'},z)\right]\\
&\quad+\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z,z'},z) - \ell(A_{T,z_i,z'},z))\ell(A_{T,z,z_i},z')\right]\\
&\quad+\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z,z_i},z') - \ell(A_{T,z',z_i},z'))\ell(A_{T,z_i,z'},z)\right]\\
&\quad= \mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_j,z'},z') - \ell(A_{T,z_j,z_i},z'))\ell(A_{T,z_j,z'},z_j)\right]\\
&\quad+\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z,z_j},z) - \ell(A_{T,z_i,z_j},z))\ell(A_{T,z,z_i},z_j)\right]\\
&\quad+\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z',z_j},z) - \ell(A_{T,z,z_j},z))\ell(A_{T,z_j,z},z')\right]\,,
\end{aligned}
$$

where in the last line we replaced z by z_j in the first term and z' by z_j in the second term and we exchanged z and z' and also z_i and z_j in the last term.

Summing I_2 and I_3 we obtain

$$
\begin{aligned}
I_2 + I_3 &= \mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z_j},z) - \ell(A_{T,z,z_j},z))(\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z,z_i},z_j))\right]\\
&+\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z,z_j},z) - \ell(A_{T,z_j,z'},z))(\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z_j,z},z'))\right]\\
&+\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z_j,z'},z'))(\ell(A_{T,z_j,z'},z) - \ell(A_{T,z_j,z'},z_j))\right]\\
&\leq 3M\mathbb{E}_{S,z}\left[\left|\ell(A_{T,z_i,z_j},z) - \ell(A_{T,z,z_j},z)\right|\right]\\
&= 3M\mathbb{E}_{S,z_i'}\left[\left|\ell(A_S,z_i) - \ell(A_{S^i},z_i)\right|\right]\,,
\end{aligned}
$$

Which proves the first part of the bound.

For the second part, we use the same technique and slightly vary the algebra. We rewrite I_2 as

$$\begin{aligned}
&\mathbb{E}_{S,z,z'}\left[\ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z_i)\right]\\
&\quad = \mathbb{E}_{S,z,z'}\left[\ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z,z_j},z')\ell(A_{T,z,z_j},z)\right]\\
&\quad \text{(renaming } z_i \text{ as } z \text{ and } z \text{ as } z' \text{ in the second term)}\\
&\quad = \mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z,z_j},z'))\ell(A_{T,z_i,z_j},z)\right]\\
&\quad +\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z_j},z) - \ell(A_{T,z,z_j},z))\ell(A_{T,z,z_j},z')\right] .
\end{aligned}$$

Next we rewrite I_3 as

$$\begin{aligned}
&\mathbb{E}_{S,z,z'}\left[\ell(A_{T,z_i,z_j},z_i)\ell(A_{T,z_i,z_j},z_j) - \ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z_i)\right]\\
&\quad = \mathbb{E}_{S,z,z'}\left[\ell(A_{T,z_i,z},z_i)\ell(A_{T,z_i,z},z) - \ell(A_{T,z_i,z_j},z)\ell(A_{T,z_i,z_j},z_i)\right]\\
&\quad \text{(renaming } z_j \text{ as } z \text{ in the first term)}\\
&\quad = \mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z},z_i) - \ell(A_{T,z_i,z_j},z_i))\ell(A_{T,z_i,z},z)\right]\\
&\quad +\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z},z) - \ell(A_{T,z_i,z_j},z))\ell(A_{T,z_i,z_j},z_i)\right]\\
&\quad = \mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z},z_i) - \ell(A_{T,z_i,z_j},z_i))\ell(A_{T,z_i,z},z)\right]\\
&\quad +\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_j,z},z) - \ell(A_{T,z_i,z_j},z))\ell(A_{T,z_i,z_j},z_j)\right]\\
&\quad \text{(exchanging } z_i \text{ and } z_j \text{ in the second term)} .
\end{aligned}$$

Summing I_2 and I_3 we obtain

$$\begin{aligned}
I_2 + I_3 \quad = \quad & \mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z_j},z') - \ell(A_{T,z,z_j},z'))\ell(A_{T,z_i,z_j},z)\right]\\
& +\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_i,z},z_i) - \ell(A_{T,z_i,z_j},z_i))\ell(A_{T,z_i,z},z)\right]\\
& +\mathbb{E}_{S,z,z'}\left[(\ell(A_{T,z_j,z},z) - \ell(A_{T,z_i,z_j},z))(\ell(A_{T,z,z_j},z') - \ell(A_{T,z_i,z_j},z_j))\right]\\
\leq \quad & M\mathbb{E}_{S,z_i',z}\left[|\ell(A_S,z) - \ell(A_{S^i},z)|\right] + M\mathbb{E}_{S,z_i'}\left[|\ell(A_S,z_j) - \ell(A_{S^i},z_j)|\right]\\
& +M\mathbb{E}_{S,z_i'}\left[|\ell(A_S,z_i) - \ell(A_{S^i},z_i)|\right] .
\end{aligned}$$

The above concludes the proof of the bound for the empirical error.

We now turn to the leave-one-out error. The bound can be obtain in a similar way. Actually, we notice that if we rewrite the derivation for the empirical error, we simply have to remove from the training set the point at which the loss is computed. That is, we simply have to replace all the quantities of the form $\ell(A_{T,z,z'},z)$ by $\ell(A_{T,z'},z)$. It is easy to see that the above results are modified in a way that gives the correct bound for the leave-one-out error.

Appendix B. Proof of Theorem 18

First we rewrite Inequality (15) in Theorem 17 as

$$\mathbb{P}_S\left[R - R_{emp}^{\gamma} > 2\frac{\beta}{\gamma} + \epsilon\left(\frac{4m\beta}{\gamma} + 1\right)\sqrt{\frac{1}{2m}}\right] \leq e^{-\epsilon^2} .$$

We introduce the following quantity

$$u(\epsilon,\gamma) = 2\frac{\beta}{\gamma} + \epsilon\left(\frac{4m\beta}{\gamma} + 1\right)\sqrt{\frac{1}{2m}} ,$$

and rewrite the above bound as

$$\mathbb{P}_S\left[R - R_{emp}^{\gamma} > u(\epsilon,\gamma)\right] \leq e^{-\epsilon^2} .$$

We define a sequence $(\gamma_k)_{k\geq 0}$ of real numbers such that

$$\gamma_k = Be^{-k}\,.$$

We define $\epsilon_k = t + \sqrt{2\ln k}$.

Now, we use the union bound to get a statement that holds for all values in the sequence $(\gamma_k)_{k\geq 1}$:

$$\begin{aligned}\mathbb{P}_S\left[\exists k\geq 1,\ R - R_{emp}^{\gamma_k} > u(\epsilon_k,\gamma_k)\right] &\leq \sum_{k\geq 1}\mathbb{P}_S\left[R - R_{emp}^{\gamma_k} > u(\epsilon_k,\gamma_k)\right]\\ &\leq \sum_{k\geq 1} e^{-\epsilon_k^2}\\ &\leq \sum_{k\geq 1}\frac{1}{k^2}e^{-t^2} \leq 2e^{-t^2}\,.\end{aligned}$$

For a given $\gamma \in (0,B]$, consider the unique value $k \geq 1$ such that $\gamma_k \leq \gamma \leq \gamma_{k-1}$. We thus have $\gamma_k \leq \gamma \leq e\gamma_k$.

The following inequalities follow from the definition of γ_k

$$\frac{1}{\gamma_k} \leq \frac{e}{\gamma}\,,$$

$$R_{emp}^{\gamma_k} \leq R_{emp}^{\gamma}\,,$$

$$\sqrt{2\ln k} = \sqrt{2\ln\ln\frac{B}{\gamma_k}} \leq \sqrt{2\ln\ln\frac{eB}{\gamma}} \equiv \alpha\,,$$

so that we have

$$u(t+\sqrt{2\ln k},\gamma_k) \leq 2\frac{e\beta}{\gamma} + (t+\alpha)\left(\frac{4me\beta}{\gamma}+1\right)\sqrt{\frac{1}{2m}} \equiv v(\gamma,t)\,.$$

We thus get the following implication

$$R - R_{emp}^{\gamma} > v(\gamma,t) \Rightarrow R - R_{emp}^{\gamma_k} > u(t+\sqrt{2\ln k},\gamma_k)\,.$$

This reasoning thus proves that

$$\mathbb{P}_S\left[\exists\gamma\in(0,B],\ R - R_{emp}^{\gamma} > v(\gamma,t)\right] \leq \mathbb{P}_S\left[\exists k\geq 0,\ R - R_{emp}^{\gamma_k} > u(t+\sqrt{2\ln k},\gamma_k)\right]\,,$$

and thus

$$\mathbb{P}_S\left[\exists\gamma\in(0,B],\ R - R_{emp}^{\gamma} > v(\gamma,t)\right] \leq 2e^{-t^2}\,,$$

which can be written as

$$\mathbb{P}_S\left[\exists\gamma\in(0,B],\ R - R_{emp}^{\gamma} > 2\frac{e\beta}{\gamma} + (t+\alpha)\left(\frac{4me\beta}{\gamma}+1\right)\sqrt{\frac{1}{2m}}\right] \leq 2e^{-t^2}\,,$$

and gives with probability $1-\delta$

$$\forall\gamma\in(0,B],\ R \leq R_{emp}^{\gamma} + 2\frac{e\beta}{\gamma} + \left(\sqrt{\ln 1/\delta} + \sqrt{2\ln\ln\frac{eB}{\gamma}}\right)\left(\frac{4me\beta}{\gamma}+1\right)\sqrt{\frac{1}{2m}}\,,$$

which gives the first inequality. The second inequality can be proven in the same way.

Appendix C. Convexity

For more details see Gordon (1999) or Rockafellar (1970). A *convex* function F is any function from a vector space $\mathcal{F}$ to $R \cup \{-\infty, +\infty\}$ which satisfies

$$\lambda F(g) + (1-\lambda)F(g') \geq F(\lambda g + (1-\lambda)g'),$$

for all $g, g' \in \mathcal{F}$ and $\lambda \in [0,1]$. A *proper* convex function is one that is always greater than $-\infty$ and not uniformly $+\infty$. The domain of F is the set of points where F is finite. A convex function is *closed* if its epigraph $\{(f,y) : y \geq F(f)\}$ is closed. The *subgradient* of a convex function at a point g, written $\partial F(g)$ is the set of vectors a such that

$$F(g') \geq F(g) + \langle g' - g, a \rangle \ ,$$

for all g'.

Convex functions are continuous on the interior of their domain and differentiable on the interior of their domain except on a set of measure zero. For a convex function F we define the *dual* of F, noted F^* by

$$F^*(a) = \sup_g \langle a, g \rangle - F(g).$$

Denoting by $\nabla F(g')$ a subgradient of F in g' (i.e. a member of $\partial F(g')$), we can define the Bregman divergence associated to F of g to g' by

$$d_F(g, g') = F(g) - F(g') - \langle g - g', \nabla F(g') \rangle \ .$$

When F is everywhere differentiable, this is well defined (since the subgradient is unique) and nonnegative (by the definition of the subgradient). Otherwise, we can define the generalized divergence as

$$\bar{d}_F(g, a) = F(g) + F^*(a) - \langle g, a \rangle \ ,$$

where $a \in \mathcal{F}^*$. Notice that this divergence is also nonnegative. Moreover, the fact that f is a minimum of F in $\mathcal{F}$ is equivalent to

$$\vec{0} \in \partial F(f),$$

which, with the following relationship

$$a \in \partial F(g) \Rightarrow F(g) + F^*(a) = \langle g, a \rangle \ ,$$

gives

$$F(f) + F^*(\vec{0}) = 0,$$

when f is a minimum of F in $\mathcal{F}$.

When F is everywhere differentiable, it is easy to get

$$\forall g \in \mathcal{F},\ d_F(g, f) = F(g) - F(f), \tag{30}$$

otherwise, using generalized divergences, we have

$$\forall g \in \mathcal{F},\ \bar{d}_F(g, \vec{0}) = F(g) - F(f). \tag{31}$$

References

N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631, 1997.

P. Bartlett For valid generalization, the size of the weights is more important than the size of the network *Advances in Neural Information Processing Systems*, 1996.

J.F. Bonnans and A. Shapiro. Optimization problems with perturbation, a guided tour. Technical Report 2872, INRIA, April 1996.

O. Bousquet and A. Elisseeff. Algorithmic stability and generalization performance. In *Neural Information Processing Systems 14*, 2001.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996a.

L. Breiman. Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24 (6):2350–2383, 1996b.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

L. Devroye. Exponential inequalities in nonparametric estimation. In *Nonparametric Functional Estimation and Related Topics*, pages 31–44. Kluwer Academic Publishers, 1991.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer Verlag, 1996.

L. Devroye and T. Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979a.

L. Devroye and T. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604, 1979b.

T. Evgeniou and M. Pontil and T. Poggio. A unified framework for Regularization Networks and Support Vector Machines. A.I. Memo 1654, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, December 1999.

G. Gordon. *Approximate Solutions to Markov Decision Processes*. PhD thesis, Carnegie Mellon University, 1999.

T. Jaakola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *Neural Information Processing Systems 12*, 1999.

M. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.

G. Lugosi and M. Pawlak. On the posterior-probability estimate of the error of nonparametric classification rules. *IEEE Transactions on Information Theory*, 40(2):475–481, 1994.

C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics*, pages 148–188. Cambridge University Press, Cambridge, 1989.

T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. In *Science*, 247(2):978–982, 1990.

R.T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

W. Rogers and T. Wagner. A finite sample distribution-free performance bound for local discrimination rules. *Annals of Statistics*, 6(3):506–514, 1978.

J.M. Steele. An Efron-Stein inequality for nonsymmetric statistics. *Annals of Statistics*, 14:753–758, 1986.

M. Talagrand. A new look at independence. *Annals of Probability*, 24:1–34, 1996.

V.N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.

G. Wahba. An introduction to model building with reproducing kernel hilbert spaces. Technical Report Statistics Department TR 1020, University of Wisconsin, Madison, 2000.