Journal of Machine Learning Research 6 (2005) 55–79 Submitted 2/04; Revised 8/04; Published 1/05

Stability of Randomized Learning Algorithms

Andre Elisseeff AEL@ZURICH.IBM.COM
IBM Zurich Research Lab
8803 Rueschlikon, Switzerland

Theodoros Evgeniou THEODOROS.EVGENIOU@INSEAD.EDU
Technology Management
INSEAD
77300 Fontainebleau, France

Massimiliano Pontil M.PONTIL@CS.UCL.AC.UK
Department of Computer Science
University College London
Gower Street, London WC1E, UK

Editor: Leslie Pack Kaelbling

Abstract

We extend existing theory on stability, namely how much changes in the training data influence the estimated models, and generalization performance of deterministic learning algorithms to the case of randomized algorithms. We give formal definitions of stability for randomized algorithms and prove non-asymptotic bounds on the difference between the empirical and expected error as well as the leave-one-out and expected error of such algorithms that depend on their random stability. The setup we develop for this purpose can be also used for generally studying randomized learning algorithms. We then use these general results to study the effects of bagging on the stability of a learning method and to prove non-asymptotic bounds on the predictive performance of bagging which have not been possible to prove with the existing theory of stability for deterministic learning algorithms.[1]

Keywords: stability, randomized learning algorithms, sensitivity analysis, bagging, bootstrap methods, generalization error, leave-one-out error.

1. Introduction

The stability of a learning algorithm, namely how changes to the training data influence the result of the algorithm, has been used by many researchers to study the generalization performance of several learning algorithms (Devroye and Wagner, 1979; Breiman, 1996b; Kearns and Ron, 1999; Bousquet and Elisseeff, 2002; Kutin and Niyogi, 2002; Poggio et al., 2004). Despite certain difficulties with theories about stability, such as the lack so far of tight bounds as well as lower bounds (Bousquet and Elisseeff, 2002), the study of learning methods using notions of stability is promising although it is still at its infancy. For example, recently Poggio et al. (2004) have shown conditions for the generalization of learning methods in terms of a stability notion that have possible implications for new insights on diverse learning problems.

[1] This work was done while A.E. was at the Max Planck Institute for Biological Cybernetics in Tuebingen, Germany.

The existing theory, however, is developed only for deterministic learning algorithms (Bousquet and Elisseeff, 2002), therefore it cannot be used to study a large number of algorithms which are randomized, such as bagging (Breiman, 1996a), neural networks, or certain Bayesian learning methods. The *goal of this paper* is to improve upon this analysis. To this end, we present a natural generalization of the existing theory to the case of randomized algorithms, thereby extending the results of (Bousquet and Elisseeff, 2002), and formally prove bounds on the performance of randomized learning algorithms using notions of randomized stability that we define. To prove our results we have also extended the results of (Bousquet and Elisseeff 2002) that hold only for symmetric learning algorithms to the case of asymmetric ones. We then prove, as an application of our results, new non-asymptotic bounds for bagging (Breiman, 1996a), a randomized learning method. Finally, we note that our work also provides an approach that can be used for extending other studies, for example other results on stability, done for deterministic algorithms to the case of randomized learning algorithms.

The paper is organized as follows. For completeness and comparison we first replicate in Section 2 the key notions of stability and the generalization bounds we extend derived for deterministic methods in the literature. We then extend these notions — Definitions 7, 10, and 13 — and generalization bounds — Theorems 9, 12 and 15 — to the case of randomized methods in Section 3. Finally, in Section 4 we present an analysis of bagging within the stability theory framework.

2. Stability and Generalization for Deterministic Algorithms

In this section we briefly review the results in (Devroye and Wagner 1979; Kearns and Ron, 1999; Bousquet and Elisseeff, 2002) that show that stability is linked to generalization for deterministic learning methods. We assume here that all algorithms are symmetric, that is, their outcome does not change when the elements in the training set are permuted. In the next section, we will extend stability concepts to the case of randomized learning methods and remove this symmetry assumption.

2.1 Basic Notation

In the following, calligraphic font is used for sets and capital letters refer to numbers unless explicitly defined. Let $\mathcal{X}$ be a set, $\mathcal{Y}$ a subset of a Hilbert space and define $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. $\mathcal{X}$ is identified as the input space and $\mathcal{Y}$ as the output space. Given a learning algorithm A we define $f_{\mathcal{D}}$ to be the solution of the algorithm when the training set $\mathcal{D} = \{z_i = (x_i, y_i),\ i = 1, \ldots, m\} \in \mathcal{Z}^m$ drawn i.i.d. from a distribution $\mathbb{P}$ is used. Algorithm A is thus interpreted as a function from $\mathcal{Z}^m$ to $(\mathcal{Y})^{\mathcal{X}}$, the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$, and we use the notation $A(\mathcal{D}) = f_{\mathcal{D}}$. We denote by $\mathcal{D}^{\setminus i}$ the training set $\mathcal{D} \setminus \{z_i\}$ obtained by removing point (x_i, y_i). More formally, point i is replaced by the empty set which we assume the learning method treats as having this point simply removed – we will need this for our analysis below. We denote by $\mathcal{D}^i$ the training set obtained by changing point (x_i, y_i) from $\mathcal{D}$ into $z' = (x', y')$, that is the set $(\mathcal{D} \setminus \{z_i\}) \cup z'$.

For any point $z = (x, y)$ and function f (real valued or binary) we denote by $\ell(f, z)$ the loss (error) when $f(x)$ is predicted instead of y (ℓ is the loss function). We define the expected error of f also known as *generalization error* by the equation

$$R_{gen}[f] = \mathbf{E}_z[\ell(f, z)].$$

We also define the *empirical error* as

$$R_{emp}\,[f] = \frac{1}{m}\sum_{i=1}^{m} \ell(f, z_i)$$

and the *leave–one–out error* as

$$R_{loo}\,[f] = \frac{1}{m}\sum_{i=1}^{m} \ell(f_{\mathcal{D}^{\setminus i}}, z_i).$$

Note that the last two errors are functions of $\mathcal{D}$. For the case of classification we use $\theta(-yf(x))$ as the loss function ℓ, where $\theta(\cdot)$ is the Heavyside function. The analysis we will do concerns classification as well as regression. For the latter we will mainly focus on the case that ℓ is a Lipschitzian loss function, that is, we assume that there exists a positive constant B such that, for every $f_1, f_2 \in (\mathcal{Y})^{\mathcal{X}}$ and $z = (x,y) \in \mathcal{Z}$, there holds the inequality $|\ell(f_1,z) - \ell(f_2,z)| \leq B|y_1 - y_2|$. Note that the absolute value satisfies this condition with $B = 1$, whereas the square loss satisfies the condition provided the set $\mathcal{Y}$ is compact.

2.2 Hypothesis Stability

The first notion of stability we consider has been stated in (Bousquet and Elisseeff, 2002) and is inspired by the work of Devroye and Wagner (1979). It is very close to what Kearns and Ron (1999) defined as hypothesis stability:

Definition 1 (Hypothesis Stability) *An algorithm A has* hypothesis stability β_m *w.r.t. the loss function ℓ if the following holds:*

$$\forall i \in \{1, \ldots, m\},\ \mathbf{E}_{\mathcal{D},z}\left[|\ell(f_{\mathcal{D}}, z) - \ell(f_{\mathcal{D}^{\setminus i}}, z)|\right] \leq \beta_m.$$

It can be shown (Bousquet and Elisseeff, 2002) that when an algorithm has hypothesis stability β_m and for *all* training sets $\mathcal{D}$ we have, for every $z \in \mathcal{Z}$, that $0 \leq \ell(f_{\mathcal{D}}, z) \leq M$, M being a positive constant, then the following relation between the leave-one-out error and the expected error holds:

Theorem 2 (Hypothesis stability leave-one-out error bound) *Let $f_{\mathcal{D}}$ be the outcome of a learning algorithm with hypothesis stability β_m (w.r.t. a loss function ℓ such that $0 \leq \ell(f,z) \leq M$). Then with probability $1 - \delta$ over the random draw of the training set $\mathcal{D}$,*

$$R_{gen}[f_{\mathcal{D}}] \leq R_{\ell oo}[f_{\mathcal{D}}] + \sqrt{\delta^{-1}\frac{M^2 + 6Mm\beta_m}{2m}}. \tag{1}$$

The proof consists of first bounding the second order moment of $(R_{gen}[f_{\mathcal{D}}] - R_{\ell oo}[f_{\mathcal{D}}])$ and then applying Chebychev's inequality. A similar bound on $(R_{gen}[f_{\mathcal{D}}] - R_{\ell oo}[f_{\mathcal{D}}])^2$ holds. Theorem 2 holds for any loss functions as long as stability can be proved w.r.t. this loss function.

In the following, we will say that an algorithm is stable when its stability scales like $1/m$, in which case the difference between the generalization and leave-one-out error is of the order $O(1/\sqrt{m})$. Many algorithms are stable according to this definition, see (Devroye et al., 1996; Bousquet and Elisseeff, 2002) for a discussion. For example, with respect to the classification loss, k-Nearest Neighbor ($k-$NN) is k/m stable. This is discussed in the next example.

Example 1 (Hypothesis Stability of k-Nearest Neighbor (k-NN)) *With respect to the classification loss, k-NN is at least $\frac{k}{m}$ stable. This can be seen via symmetrization arguments. For the sake of simplicity we give here the proof for the* 1*-NN only. Let v_i be the neighborhood of z_i such that the closest point in the training set to any point of v_i is z_i. The $1-$NN algorithm computes its output via the following equation (we assume here that the probability that x_i appears twice in the training set is negligible):*

$$f_{\mathcal{D}}(x) = \sum_{i=1}^{m} y_i \mathbf{1}_{x \in v_i}(x)$$

where $\mathbf{1}_S$ is the indicator function of set S. The difference between the losses $\ell(f_{\mathcal{D}}, z)$ and $\ell(f_{\mathcal{D}^{\setminus i}}, z)$ is then defined by the set v_i. Here we assume that ℓ is the classification loss. We then have that

$$\mathbf{E}_z[|\ell(f_{\mathcal{D}_m}, z) - \ell(f_{\mathcal{D}^{\setminus i}}, z)|] \leq \mathbb{P}(v_i).$$

Note that v_i depends on $\mathcal{D}$. Now averaging over $\mathcal{D}$ we need to compute $\mathbf{E}_{\mathcal{D}}[\mathbb{P}(v_i)]$ which is the same for all i because the z_i are drawn i.i.d. from the same distribution. But, we have,

$$1 = \mathbf{E}_{\mathcal{D},z}\left[|f_{\mathcal{D}}(x)|\right] = \mathbf{E}_{\mathcal{D},z}\left[\left|\sum_{i=1}^{m} y_i \mathbf{1}_{x \in v_i}(x)\right|\right] = \mathbf{E}_{\mathcal{D},z}\left[\sum_{i=1}^{m} \mathbf{1}_{x \in v_i}(x)\right].$$

The last equality comes from the fact that for fixed $\mathcal{D}$ and z, only one $\mathbf{1}_{x \in v_i}(x)$ is non-zero. We also have that

$$1 = \mathbf{E}_{\mathcal{D},z}\left[\sum_{i=1}^{m} \mathbf{1}_{x \in v_i}(x)\right] = m\mathbf{E}_{\mathcal{D}}\left[\mathbb{P}(v_i)\right].$$

Consequently, $\mathbf{E}_{\mathcal{D}}[\mathbb{P}(v_i)] = \frac{1}{m}$ and the 1*-NN has hypothesis stability bounded above by $1/m$.*

A bound similar to Equation (1) can be derived for the empirical error when a slightly different notion of stability is used (Bousquet and Elisseeff, 2002).[2]

Definition 3 (Pointwise hypothesis stability) *An algorithm A has* pointwise hypothesis stability *β_m w.r.t. the loss function ℓ if the following holds :*

$$\forall i \in \{1, \dots, m\},\ \mathbf{E}_{\mathcal{D},z}\left[\left|\ell(f_{\mathcal{D}}, z_i) - \ell(f_{\mathcal{D}^{\setminus i} \cup z}, z_i)\right|\right] \leq \beta_m.$$

Note that we adopted the same notation β_m for all notions of stability since it should always be clear from the context which is the referred notion. As for the case of hypothesis stability and leave-one-out error above, it can also be shown (Bousquet and Elisseeff, 2002) that when an algorithm has pointwise hypothesis stability β_m and if for all training sets $\mathcal{D}$, $0 \leq \ell(f, z) \leq M$, then the following relation between the empirical error and the expected error holds:

Theorem 4 (Pointwise hypothesis stability empirical error bound) *Let $f_{\mathcal{D}}$ be the outcome of a learning algorithm with pointwise hypothesis stability β_m (w.r.t. a loss function ℓ such that $0 \leq \ell(f_{\mathcal{D}}, z) \leq M$). Then with probability $1 - \delta$ over the random draw of the training set $\mathcal{D}$,*

$$R_{gen}[f_{\mathcal{D}}] \leq R_{emp}[f_{\mathcal{D}}] + \sqrt{\delta^{-1} \frac{M^2 + 12Mm\beta_m}{2m}}. \qquad (2)$$

[2] We slightly changed the definition to correct one mistake that has been pointed out by Poggio et al., (2004): the difference of losses is taken here between two outcomes trained on datasets of equal sizes.

2.3 Uniform Stability

The application of bound (1) to different algorithms $f_1, \dots, f_Q$ with stabilities β_m^q, $q = 1, \dots, Q$, is usually done by using the union bound (Vapnik, 1998). Applying Theorem 2 Q times, we get with probability $1 - \delta$,

$$\forall q \in \{1, \dots, Q\}, \quad R_{gen}[f_q] \leq R_{\ell oo}[f_q] + \sqrt{\delta^{-1} Q \frac{M^2 + 6Mm\beta_m^q}{2m}}. \tag{3}$$

In such situations, we would like to have a dependence in $\log(Q)$ so that we can have large values of Q without increasing the bound too much. To this end, we need a stronger notion of stability called uniform stability (Bousquet and Elisseeff, 2002).

Definition 5 (Uniform Stability) *An algorithm A has* uniform stability β_m *w.r.t. the loss function ℓ if the following holds*

$$\forall \mathcal{D} \in \mathcal{Z}^m, \ \forall i \in \{1, \dots, m\}, \ \|\ell(f_\mathcal{D}, .) - \ell(f_{\mathcal{D}^{\setminus i}}, .)\|_\infty \leq \beta_m. \tag{4}$$

It is easily seen that the uniform stability is an upper bound on hypothesis and pointwise hypothesis stability (Bousquet and Elisseeff, 2002). Uniform stability can be used in the context of regression to get bounds as follows (Bousquet and Elisseeff, 2002):

Theorem 6 *Let $f_\mathcal{D}$ be the outcome of an algorithm with uniform stability β_m w.r.t. a loss function ℓ such that $0 \leq \ell(f_\mathcal{D}, z) \leq M$, for all $z \in \mathcal{Z}$ and all sets $\mathcal{D}$. Then, for any $m \geq 1$, and any $\delta \in (0,1)$, each of the following bounds holds with probability $1 - \delta$ over the random draw of the training set $\mathcal{D}$,*

$$R_{gen}[f_\mathcal{D}] \leq R_{emp}[f_\mathcal{D}] + 2\beta_m + (4m\beta_m + M)\sqrt{\frac{\log(1/\delta)}{2m}}, \tag{5}$$

and

$$R_{gen}[f_\mathcal{D}] \leq R_{\ell oo}[f_\mathcal{D}] + \beta_m + (4m\beta_m + M)\sqrt{\frac{\log(1/\delta)}{2m}}. \tag{6}$$

The dependence on δ is $\sqrt{\log(1/\delta)}$ which is better than the bounds given in terms of hypothesis and pointwise hypothesis stability.

It is important to note that these bounds hold only for regression. Uniform stability can also be used for classification with margin classifiers to get similar bounds, but we do not pursue this here for simplicity. In the next section, for simplicity we also consider random uniform stability only for regression. Classification can be treated with appropriate changes like in (Bousquet and Elisseeff, 2002).

Example 2 (Uniform Stability of regularization methods) *Regularization-based learning algorithms such as Regularization Networks (RN's) (Poggio and Girosi, 1990) and Support Vector Machines (SVM's), see, for example, (Vapnik, 1998), are obtained by minimizing the functional*

$$\sum_{i=1}^{m} \ell(f, z_i) + \lambda \|f\|_K^2$$

where $\lambda > 0$ is a regularization parameter and $\|f\|_K$ is the norm of f in a reproducing kernel Hilbert space associated to a symmetric and positive definite kernel $K : X \times X \to \mathbb{R}$. A typical example is the Gaussian, $K(x,t) = \exp(-\|x-t\|^2/2\sigma^2)$, where σ is a parameter controlling the width of the kernel. Depending on the loss function used, we obtain different learning methods. RN's use the square loss while SVM's regression uses the loss $\ell(f,z) = |f(x)-y|_\varepsilon$, where $|\xi|_\varepsilon = |\xi| - \varepsilon$ if $|\xi| > \varepsilon$, and zero otherwise.[3]

It can be shown (Bousquet and Elisseeff, 2002) that for Lipschitz loss functions, the uniform stability of these regularization methods scales as $1/\lambda$. This results is in agreement with the fact that for small λ, the solution tends to fit perfectly the data and Theorem 6 does not give an interesting bound. On the contrary, when λ is large the solution is more stable and Theorem 6 gives a tight bound. Hence, there is a trade-off between stability and deviation between generalization and empirical error that is illustrated here by the role of the regularization parameter λ.

Finally, we note that the notion of uniform stability may appear a little restrictive since the inequality in Equation (4) has to hold over all training sets $\mathcal{D}$. A weaker notion of stability has been introduced by Kutin and Niyogi (2002) with related exponential bounds. We do not discuss this issue here for simplicity, and we conjecture that the analysis we do below can be generally adapted for other notions of stability.

3. Stability and Generalization for Randomized Algorithms

The results summarized in the previous section concern only deterministic learning algorithms. For example they cannot be applied to certain neural networks as well as bagging methods. In this section we generalize the theory to include randomized learning algorithms.

3.1 Informal Reasoning

Let A be a randomized learning algorithm, that is a function from $\mathcal{Z}^m \times \mathcal{R}$ onto $(\mathcal{Y})^{\mathcal{X}}$ where $\mathcal{R}$ is a space containing elements $\mathbf{r}$ that model the randomization of the algorithm and is endowed with a probability measure $\mathbb{P}_\mathbf{r}$. For notational convenience, we will use the shorthand $f_{\mathcal{D},\mathbf{r}}$ to denote the outcome of the algorithm A applied on a training set $\mathcal{D}$ with a random parameter $\mathbf{r}$. We should distinguish between two types of randomness that are exemplified by the following examples.

Example 3 (Bootstrapping once) *Let $\mathcal{R} = \{1,\ldots,m\}^p$, $p \leq m$, and define $\mathbb{P}_\mathbf{r}$, for $\mathbf{r} \in \mathcal{R}$, to be a multinomial distribution with m parameters $(1/m,\ldots,1/m)$. This random process models the sub-sampling with replacement of p elements from a set of m distinct elements. An algorithm A that takes as input a training set $\mathcal{D}$, performs a sub-sampling with replacement and runs a method such as a decision tree on the sub-sampled training set is typically modeled as a randomized algorithm taking as inputs a training set and an element $\mathbf{r} \in \mathcal{R}$ just described.*

In this first example we see that the randomness depends on m, which is different from what the second example describes.

[3] Note that in the statistical learning theory literature (Vapnik, 1998), SVM are usually presented in term of mathematical programming problems and the parameter λ is replaced by $C = 1/(2\lambda)$ which now appears in front of the empirical error.

Example 4 (Initialization weights) *Let $\mathcal{R} = [0,1]^k$ and define $\mathbb{P}_{\mathbf{r}}$ to be the uniform distribution over $\mathcal{R}$. Such a random process appear in the initialization procedure of Neural Networks when the initial weights are chosen randomly. In the latter case, a multi-layer perceptron with k weights can be understood as an algorithm A taking a training set and a random vector $\mathbf{r} \in \mathcal{R}$ as inputs.*

We consider the following issues for the definitions of stability for randomized algorithms below.

- We give stability definitions that reduce to deterministic stability concepts when there is no randomness, that is, $\mathcal{R}$ is reduced to one element with probability 1.

- We assume that the randomness of an algorithm (randomness of $\mathbf{r}$) is independent of the training set $\mathcal{D}$, although $\mathbf{r}$ may depend on the size of this set, m. There are two main reasons for this: first, it simplifies the calculations; second, the randomness of $\mathbf{r}$ has generally nothing to do with the randomness of the training set $\mathcal{D}$. Most of the time our knowledge about the distribution over $\mathbf{r}$ is known perfectly, like in the examples above, and we would like to take advantage of that. Adding some dependencies between $\mathbf{r}$ and $\mathcal{D}$ reduces this knowledge since nothing is assumed about the distribution over $\mathcal{Z}$ from which $\mathcal{D}$ is drawn.

- We also consider the general case that the randomization parameter $\mathbf{r} \in \mathcal{R}^T$ is decomposed as a vector of independent random parameters $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_T)$ where each $\mathbf{r}_t$ is drawn from the distribution $\mathbb{P}^t_{\mathbf{r}_t}$. For example, this model can be used to model the randomization of bagging (Breiman, 1996a), where each $\mathbf{r}_t$ corresponds to one random subsampling from the data, and the T subsamples are all drawn independently. To summarize, we will make use of the following assumption:

  Assumption 1: *We assume that $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_T)$ where $\mathbf{r}_t$, $t = 1, \ldots, T$ are random elements drawn independently from the same distribution and write $\mathbf{r} \in \mathcal{R}^T$ to indicate the product nature of* $\mathbf{r}$.

- Finally we assume that we can re-use a draw of $\mathbf{r}$ for different training set sizes, for example for m and $m-1$. We need this assumption for the definitions of stability below to be well defined as well as for the leave-one-out error definition we use for randomized methods.

To develop the last issue further, let us consider how to compute a leave-one-out error estimate when the algorithm depends on a random vector $\mathbf{r}$ that changes with the number of training examples. One way is to sample a new random vector $\mathbf{r}$ (which in this case will concern only $m-1$ training points) for each fold/iteration. This is done, for example, by Kearns and Ron (1999) when they introduce the notion of the random error stability. However, this introduces more instabilities to the algorithms whose behavior can be different not only because of changes in the training set but also because of changes in the random part $\mathbf{r}$. A more stable leave-one-out procedure for a randomized algorithm would be to fix $\mathbf{r}$ and to apply the leave-one-out method only on the sampling of the training set – a deterministic leave-one-out error (Evgeniou et al., 2004). Therefore for each leave-one-out iteration, when we leave one point out — which is replaced, as we discussed in Section 2.1, with an empty set which we assume the learning method does not use — we use the same $\mathbf{r}$ for the remaining $m-1$ points. For instance, in Example 3.1 we would use the same bootstrap samples that we used when having all m points, with the point left out replaced by the empty set that is not used for training, for each leave-one-out iteration. In that case, we don't need to re-sample $\mathbf{r}$ and the leave-one-out estimate concerns an algorithm that is closer to what we consider on m points.

Therefore, in what follows, keeping in mind Example 3, we assume the following:

Assumption 2: *The same* $\mathbf{r}$ *can be applied to* $f_{\mathcal{D}}$ *and* $f_{\mathcal{D}^{\setminus i}}$ *where* $\mathcal{D}^{\setminus i}$ *is the set* $\mathcal{D}$ *where point* i *is replaced by the empty set. We also consider the deterministic leave-one-out error computed as described above.*

Note that this assumption is not restrictive about the kind of learning methods we can consider. For example both in Example 3.1 and 3.2 the same $\mathbf{r}$ (i.e. subsamples or initialization of neural network weights) can be used for m and $m-1$ training points.

3.2 Random Hypothesis Stability

The first definition we consider is inspired by the hypothesis stability for deterministic algorithms.

Definition 7 (Random Hypothesis Stability) *A randomized algorithm* A *has* random hypothesis stability β_m *w.r.t. the loss function* ℓ *if the following holds:*

$$\forall i \in \{1,\ldots,m\}, \mathbf{E}_{\mathcal{D},z,\mathbf{r}}\left[\left|\ell(f_{\mathcal{D},\mathbf{r}},z)-\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z)\right|\right] \leq \beta_m. \tag{7}$$

Note that the value in the left hand side (l.h.s.) of Equation (7) can vary for different indexes i. If $\mathbf{r}$ is fixed then the random hypothesis stability is exactly the same as the hypothesis stability except that the resulting algorithm need not be symmetric anymore: if we sample the training data using a fixed $\mathbf{r}$, permuting two data points might lead to different samplings and hence to a different outcome. This means that we cannot apply the results for the case of deterministic algorithms and we have to consider other bounds on the variance of the difference between the generalization and empirical (or leave-one-out) errors. We prove in the appendix the following lemma.

Lemma 8 *For any (non-symmetric) learning algorithm* A *and loss function* ℓ *such that* $0 \leq \ell(f,z) \leq M$ *we have for the leave-one-out error:*

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen}-R_{\ell oo})^2\right] \leq \frac{2M^2}{m}+\frac{12M}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[|\ell(f_{\mathcal{D}},z)-\ell(f_{\mathcal{D}^{\setminus i}},z)|\right]. \tag{8}$$

Using Chebychev's inequality, this lemma leads to the inequality

$$\mathbb{P}_{\mathcal{D}}\left(R_{gen}[f_{\mathcal{D},\mathbf{r}}]-R_{\ell oo}[f_{\mathcal{D},\mathbf{r}}] \geq \varepsilon \mid \mathbf{r}\right) \leq \frac{2M^2}{m\varepsilon^2}+\frac{12M\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(f_{\mathcal{D},\mathbf{r}},z)-\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z)\right|,\mathbf{r}\right]}{m\varepsilon^2}, \tag{9}$$

where we use the notation $\mathbb{E}[X,Y]$ for the expectation of X conditioned on Y, and $\mathbb{P}[.|\mathbf{r}]$ for the conditional probability. By integrating Equation (9) with respect to $\mathbf{r}$ and using the property $\mathbb{E}_Y\left[\mathbb{E}_X[g(X,Y)|Y]\right] = \mathbb{E}_{X,Y}[g(X,Y)]$ we derive the following theorem about the generalization and leave-one-out errors of randomized learning methods:

Theorem 9 *Let $f_{\mathcal{D},\mathbf{r}}$ be the outcome of a randomized algorithm with random hypothesis stability β_m w.r.t. a loss function ℓ such that $0 \le \ell(f,z) \le M$, for all $y \in \mathcal{Y}$, $\mathbf{r} \in \mathcal{R}$ and all sets $\mathcal{D}$. Then with probability $1-\delta$ with respect to the random draw of the $\mathcal{D}$ and $\mathbf{r}$,*

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \le R_{\ell oo}[f_{\mathcal{D},\mathbf{r}}] + \sqrt{\delta^{-1}\frac{2M^2 + 12Mm\beta_m}{m}}. \tag{10}$$

Notice that in the case that we make Assumption 1 nothing changes since the integration of (9) w.r.t. $\mathbf{r}$ does not depend on the decomposition nature of $\mathbf{r}$ made in Assumption 1.

As in the deterministic case, it is possible to define a different notion of stability to derive bounds on the deviation between the empirical error and the generalization error of randomized algorithms:

Definition 10 (Random Pointwise Hypothesis Stability) *A randomized algorithm A has* random pointwise hypothesis stability β_m *w.r.t. the loss function ℓ if the following holds:*

$$\forall i \in \{1,\dots,m\}, E_{\mathcal{D}_m,\mathbf{r},z}\left|\ell(f_{\mathcal{D},\mathbf{r}},z_i) - \ell(f_{\mathcal{D}^{\setminus i}\cup z,\mathbf{r}},z_i)\right| \le \beta_m. \tag{11}$$

Using the following lemma proved in the appendix,

Lemma 11 *For any (non-symmetric) learning algorithm A and loss function ℓ such that $0 \le \ell(f,z) \le M$ we have for the empirical error,*

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen} - R_{emp})^2\right] \le \frac{2M^2}{m} + \frac{12M}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[|\ell(f_{\mathcal{D}},z_i) - \ell(f_{\mathcal{D}^{\setminus i}\cup z},z_i)|\right]. \tag{12}$$

we can derive as before the theorem:

Theorem 12 *Let $f_{\mathcal{D},\mathbf{r}}$ be the outcome of a random algorithm with random pointwise hypothesis stability β_m w.r.t. a loss function ℓ such that $0 \le \ell(f,z) \le M$, for all $y \in \mathcal{Y}$, $\mathbf{r} \in \mathcal{R}$ and all sets $\mathcal{D}$. Then with probability $1-\delta$ with respect to the random draw of the $\mathcal{D}$ and $\mathbf{r}$,*

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \le R_{emp}[f_{\mathcal{D},\mathbf{r}}] + \sqrt{\delta^{-1}\frac{2M^2 + 12Mm\beta_m}{m}}. \tag{13}$$

We note that both for Theorems 9 and 12 (Lemmas 8 and 11) one can further improve the constants of the bounds – as is typically the case with bounds in the literature.

The parallel with the deterministic case is striking. However when we consider a random space $\mathcal{R}$ reduced to only one element, then the bounds we obtain here are worse since we assume non-symmetric learning algorithms.

3.3 Random Uniform Stability

The uniform stability definition (Definition 5) for deterministic algorithms can be extended as follows:

Definition 13 (Uniform Stability of Randomized Algorithms) *We say that a randomized learning algorithm has uniform stability β_m w.r.t. the loss function ℓ if, for every $i = 1, \ldots, m$*

$$\sup_{\mathcal{D},z} \left| \mathbf{E}_{\mathbf{r}} \left[\ell(f_{\mathcal{D},\mathbf{r}}, z) \right] - \mathbf{E}_{\mathbf{r}} \left[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}}, z) \right] \right| \leq \beta_m. \tag{14}$$

Note that this definition is consistent with Definition 5 which holds for deterministic symmetric learning algorithms.

To link uniform stability to generalization, the following result by McDiarmid (1989), see also (Devroye et al., 1996), is central.

Theorem 14 (Bounded Difference Inequality) *Let $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_T) \in \mathcal{R}$ be T independent random variables ($\mathbf{r}_t$ can be vectors, as in Assumption 1, or scalars) drawn from the same probability distribution $\mathbb{P}_{\mathbf{r}}$. Assume that the function $G : \mathcal{R}^T \to \mathbb{R}$ satisfies*

$$\sup_{\mathbf{r}_1,\ldots,\mathbf{r}_T,\mathbf{r}'_t} \left| G(\mathbf{r}_1, \ldots, \mathbf{r}_T) - G(\mathbf{r}_1, \ldots, \mathbf{r}_{t-1}, \mathbf{r}'_t, \mathbf{r}_{t+1}, \ldots, \mathbf{r}_T) \right| \leq c_t, \; t = 1, \ldots, T. \tag{15}$$

where c_t is a nonnegative function of t. Then, for every $\varepsilon > 0$

$$\mathbb{P}\left[G(\mathbf{r}_1, \ldots, \mathbf{r}_T) - \mathbf{E}_{\mathbf{r}}\left[G(\mathbf{r}_1, \ldots, \mathbf{r}_T) \right] \geq \varepsilon \right] \leq \exp\left\{ -2\varepsilon^2 / \sum_{t=1}^{T} c_t^2 \right\}. \tag{16}$$

For the next theorem we replace the G of Theorem 14 with $\ell(f_{\mathcal{D},\mathbf{r}}, z)$ and require that, for every $\mathcal{D} \in \mathcal{Z}^m$ and $z \in \mathcal{Z}$, $\ell(f_{\mathcal{D},\mathbf{r}}, z)$ satisfies the inequality in Equation (15). This is a mild assumption but the bounds below will be interesting only if, for $T \to \infty$, c_t goes to zero at least as $1/\sqrt{T}$. We use ρ as the supremum of the c_ts of Theorem 14.

Theorem 15 *Let $f_{\mathcal{D},\mathbf{r}}$ be the outcome of a randomized learning algorithm satisfying Assumptions 1 and 2 with uniform stability β_m w.r.t. the loss function ℓ. Let ρ be such that for all t*

$$\sup_{\mathbf{r}_1,\ldots,\mathbf{r}_T,\mathbf{r}'_t} \sup_{z} \left| \ell(f_{\mathcal{D},(\mathbf{r}_1,\ldots,\mathbf{r}_T)}, z) - \ell(f_{\mathcal{D},(\mathbf{r}_1,\ldots,\mathbf{r}_{t-1},\mathbf{r}'_t,\mathbf{r}_{t+1},\ldots,\mathbf{r}_T)}, z) \right| \leq \rho,$$

as in Equation (15) for G being $\ell(f_{\mathcal{D},\mathbf{r}}, z)$ and $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_T)$. The following bound holds with probability at least $1 - \delta$ with respect to the random draw of the $\mathcal{D}$ and $\mathbf{r}$,

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \leq R_{emp}(f_{\mathcal{D},\mathbf{r}}) + 2\beta_m + \left(\frac{M + 4m\beta_m}{\sqrt{2m}} + \sqrt{2T}\rho \right) \left(\sqrt{\log 2/\delta} \right), \tag{17}$$

and,

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \leq R_{\ell oo}(f_{\mathcal{D},\mathbf{r}}) + \beta_m + \left(\frac{M + 2m\beta_{m-1} + 2m\beta_m}{\sqrt{2m}} + \sqrt{2T}\rho \right) \left(\sqrt{\log(2/\delta)} \right). \tag{18}$$

Furthermore, assuming that β_{m-1}, the random uniform stability for training sets of size $m-1$, is greater than β_m, we can simplify Equation (18) to:

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \leq R_{\ell oo}(f_{\mathcal{D},\mathbf{r}}) + \beta_m + \left(\frac{M + 4m\beta_{m-1}}{\sqrt{2m}} + \sqrt{2T}\rho\right)(\sqrt{\log(2/\delta)}). \tag{19}$$

Notice that the assumption for the simplification we make in the theorem that β_{m-1} is greater than β_m is natural: when points are added to the training set, the outcome of a learning algorithm is usually more stable. Moreover, bounds on β_m can be used here so that the condition $\beta_{m-1} \geq \beta_m$ can be replaced by a condition on these bounds: we would require that the bounds on β_m are non-increasing in m.

We note that ρ may depend both on the number of random variables T and the number of training data m. In the bagging example below we estimate a bound on ρ that depends only on T, the number of subsamples we do for the bagging process – it may or may not be possible to show that ρ depends on m, too, but this is an open question. We do not know of an example where ρ also depends on m or, alternatively, of a case where it can be shown that it is not possible to have ρ depend on m. The latter case would imply that for fixed T the empirical (leave-one-out) error does not converge to the expected error as m increases. This is, however, an open question and potentially a weakness for the framework we develop here.

Finally note that, as in the deterministic case discussed in Section 2, results similar to those in Theorem 15 can be given for classification following the same line as in (Bousquet and Elisseeff, 2002).

4. Stability of Bagging and Subbagging

In this section we discuss an application of the results derived above to bagging (Breiman, 1996a) and subbagging, see, for example, (Andonova et al., 2002), two randomized algorithms which work by averaging the solutions of a learning algorithm trained a number of times on random subsets of the training set. We will analyze these methods within the stability framework presented above. To this end, we need to study how bagging and subbagging “affect” the stability of the base (underlying) learning algorithm. First we present more formally what we mean by bagging.

4.1 Bagging

Bagging consists of training the same learning algorithm on a number T of different bootstrap sets of a training set $\mathcal{D}$ and by averaging the obtained solutions. We denote these bootstrap sets by $\mathcal{D}(\mathbf{r}_t)$ for $t = 1,\ldots,T$, where the $\mathbf{r}_t \in \mathcal{R} = \{1,\ldots,m\}^m$ are instances of a random variable corresponding to sampling *with* replacement of m elements from the training set $\mathcal{D}$ (recall the notation in Example 3). Such random variables have a multinomial distribution with parameters $(\frac{1}{m},\ldots,\frac{1}{m})$. The overall bagging model can thus be written as:

$$F_{\mathcal{D},\mathbf{r}} = \frac{1}{T}\sum_{t=1}^{T} f_{\mathcal{D}(\mathbf{r}_t)}. \tag{20}$$

Here we assume that the base learning method ($f_{\mathcal{D}}$) treats multiple copies of a training point (for example when many copies of the same point are sampled) as one point.[4] Extending the results below to the case where multiple copies of a point are treated as such is an open question.

The reader should also keep in mind that the base learning algorithm may be itself randomized with random parameter $\mathbf{s}$. When trained on the $t-$th bootstrap set, $\mathcal{D}(\mathbf{r}_t)$, this algorithm will output the solution $f_{\mathcal{D}(\mathbf{r}_t),\mathbf{s}_t}$. However, to simplify the notation, we suppress the symbol $\mathbf{s}_t$ in our discussion below.

In what follows, we compute an upper bound on the random hypothesis stability for bagging. For regression, we have then the following proposition:

Proposition 4.1 (Random hypothesis stability of bagging for regression) *Assume that the loss ℓ is $B-$lipschitzian w.r.t. its first variable. Let $F_{\mathcal{D},\mathbf{r}}$, $\mathbf{r} \in \mathcal{R}^T$, be the outcome of a bagging algorithm whose base machine ($f_{\mathcal{D}}$) has (pointwise) hypothesis stability γ_m w.r.t. the ℓ_1 loss function. Then the random (pointwise) hypothesis stability β_m of $F_{\mathcal{D},\mathbf{r}}$ with respect to ℓ is bounded by*

$$\beta_m \leq B \sum_{k=1}^{m} \frac{k\gamma_k}{m} \mathbb{P}_{\mathbf{r}}\left[d(\mathbf{r}) = k\right],$$

where $d(\mathbf{r})$, $\mathbf{r} \in \mathcal{R}$, is the number of distinct sampled points in one bootstrap iteration.

Proof

We first focus on hypothesis stability. Let us assume first that $\mathcal{D}$ is fixed and z too. We would like to bound:

$$I(\mathcal{D},z) = \mathbf{E}_{\mathbf{r}_1,\ldots,\mathbf{r}_T}\left[\left|\ell\left(\frac{1}{T}\sum_{t=1}^{T} f_{\mathcal{D}(\mathbf{r}_t)}, z\right) - \ell\left(\frac{1}{T}\sum_{t=1}^{T} f_{\mathcal{D}^{\setminus i}(\mathbf{r}_t)}, z\right)\right|\right]$$

where $\mathbf{r}_1,\ldots,\mathbf{r}_T$ are i.i.d. random variables modeling the random sampling of bagging and having the same distribution as $\mathbf{r}$. Since ℓ is $B-$lipschitzian, and the $\mathbf{r}_t$ are i.i.d., $I(\mathcal{D},z)$ can be bounded as:

$$\begin{aligned} I(\mathcal{D},z) &\leq \frac{B}{T}\,\mathbf{E}_{\mathbf{r}_1,\ldots,\mathbf{r}_T}\left[\left|\sum_{t=1}^{T}\left(f_{\mathcal{D}(\mathbf{r}_t)}(x) - f_{\mathcal{D}^{\setminus i}(\mathbf{r}_t)}(x)\right)\right|\right] \\ &\leq \frac{B}{T}\sum_{t=1}^{T}\mathbf{E}_{\mathbf{r}_t}\left[\left|f_{\mathcal{D}(\mathbf{r}_t)}(x) - f_{\mathcal{D}^{\setminus i}(\mathbf{r}_t)}(x)\right|\right] = B\,\mathbf{E}_{\mathbf{r}}\left[\left|f_{\mathcal{D}(\mathbf{r})}(x) - f_{\mathcal{D}^{\setminus i}(\mathbf{r})}(x)\right|\right]. \end{aligned}$$

To simplify the notation we denote by $\Delta(\mathcal{D}(\mathbf{r}),x)$ the difference between $f_{\mathcal{D}^{\setminus i}(\mathbf{r})}(x)$ and $f_{\mathcal{D}(\mathbf{r})}(x)$. We have that

$$\begin{aligned} \mathbf{E}_{\mathbf{r}}\left[|\Delta(\mathcal{D}(\mathbf{r}),x)|\right] &= \mathbf{E}_{\mathbf{r}}\left[|\Delta(\mathcal{D}(\mathbf{r}),x)|\left(\mathbf{1}_{i\in\mathbf{r}} + \mathbf{1}_{i\notin\mathbf{r}}\right)\right] \\ &= \mathbf{E}_{\mathbf{r}}\left[|\Delta(\mathcal{D}(\mathbf{r}),x)|\,\mathbf{1}_{i\in\mathbf{r}}\right] + \mathbf{E}_{\mathbf{r}}\left[|\Delta(\mathcal{D}(\mathbf{r}),x)|\,\mathbf{1}_{i\notin\mathbf{r}}\right]. \end{aligned}$$

Note that the second part of the last line is equal to zero because when i is not in $\mathbf{r}$, point z_i does not belong to $\mathcal{D}(\mathbf{r})$ and, thus, $\mathcal{D}(\mathbf{r}) = \mathcal{D}^{\setminus i}(\mathbf{r})$. We conclude that

$$I(\mathcal{D},z) \leq B\mathbf{E}_{\mathbf{r}}\left[|\Delta(\mathcal{D}(\mathbf{r}),x)|\,\mathbf{1}_{i\in\mathbf{r}}\right].$$

[4] This means that if for example the underlying learning algorithm is a neural network, this algorithm is modified by a preprocessing step so that the training set consists only of distinct data points.

We now take the average w.r.t. $\mathcal{D}$ and z:

$$\mathbf{E}_{\mathcal{D},z}\left[I(\mathcal{D},z)\right] \leq B\mathbf{E}_{\mathbf{r},\mathcal{D},x}\left[|\Delta(\mathcal{D}(\mathbf{r}),x)|\mathbf{1}_{i\in\mathbf{r}}\right] =$$
$$= B\mathbf{E}_{\mathbf{r}}\left[\mathbf{E}_{\mathcal{D},x}\left[|\Delta(\mathcal{D}(\mathbf{r}),x)|\right]\mathbf{1}_{i\in\mathbf{r}}\right] = B\mathbf{E}_{\mathbf{r}}\left[\gamma_{d(\mathbf{r})}\mathbf{1}_{i\in\mathbf{r}}\right], \tag{21}$$

where the last equality follows by noting that $\mathbf{E}_{\mathcal{D},x}\left[|\Delta(\mathcal{D}(\mathbf{r}),x)|\right]$ is bounded by the hypothesis stability $\gamma_{d(\mathbf{r})}$ of a training set of size $d(\mathbf{r})$. We now note that when averaging w.r.t. $\mathbf{r}$, the important variable about $\mathbf{r}$ is the size $d(\mathbf{r})$:

$$\mathbf{E}_{\mathbf{r}}\left[\gamma_{d(\mathbf{r})}\mathbf{1}_{i\in\mathbf{r}}\right] = \sum_{k=1}^{m}\mathbb{P}_{\mathbf{r}}\left[d(\mathbf{r})=k\right]\gamma_k\mathbf{E}_{\mathbf{r}}\left[\mathbf{1}_{i\in\mathbf{r}};d(\mathbf{r})=k\right].$$

Now note that, by symmetry, $\mathbf{E}_{\mathbf{r}}\left[\mathbf{1}_{i\in\mathbf{r}};d(\mathbf{r})=k\right]=k/m$. This concludes the proof for hypothesis stability. The proof for pointwise stability is exactly the same except that in Equation (21) there is no expectation w.r.t. z and z is replaced by z_i. ■

The bounds we just proved depend on the quantities $\mathbb{P}_{\mathbf{r}}[d(\mathbf{r})=k]$, where, we recall that $d(\mathbf{r})$, $\mathbf{r}\in\mathcal{R}$, is the number of distinct sampled points in one bootstrap iteration. It can be shown, for example by applying Theorem 14, that the random variable $d(\mathbf{r})$ is sharply concentrated around its mode which is for $k=(1-\frac{1}{e})m\approx 0.632m$. For that reason, in what follows we will assume that the previous bounds can be approximately rewritten as:

$$\beta_m \leq .632B\gamma_{.632m}.$$

For example if $B=1$ and γ_m scales appropriately with m the bounds on the random (pointwise) hypothesis stability of the bagging predictor are better than those on the (pointwise) hypothesis stability of a single predictor trained on the whole training set. Notice also that .632 is the probability that the bootstrapped set will contain a specific (any) point, also used to justify the .632 bootstrap error estimates (Efron and Tibshirani, 1997).

Similar results can be shown for the random (pointwise) hypothesis stability for classification. In particular:

Proposition 4.2 (Random hypothesis stability of bagging for classification) *Let $F_{\mathcal{D},\mathbf{r}}$ be the outcome of a bagging algorithm whose base machine has (pointwise) hypothesis stability γ_m w.r.t. the classification loss function. Then, the (pointwise) random hypothesis stability β_m of $F_{\mathcal{D},\mathbf{r}}$ w.r.t. the ℓ_1 loss function is bounded by*

$$\beta_m \leq 2\sum_{k=1}^{m}\frac{k\gamma_k}{m}\mathbb{P}_{\mathbf{r}}\left[d(\mathbf{r})=k\right].$$

Proof The proof is the same as in the above proposition except that the loss appearing therein is the ℓ_1 loss and, so, $B=1$. The functions $f^{(t)}$ being $\{+1,-1\}$ valued, the term:

$$\mathbf{E}_{\mathcal{D},z}\left[|f_{\mathcal{D}}(x)-f_{\mathcal{D}^{\setminus i}}(x)|\right]$$

is equal to the term

$$2\mathbf{E}_{\mathcal{D},z}\left[\theta(-yf_{\mathcal{D}}(x))-\theta(-yf_{\mathcal{D}^{\setminus i}}(x))\right].$$

So that stability w.r.t. the ℓ_1 loss function can be replaced by stability w.r.t. the classification loss, and the proof can be transposed directly. ■

Example 5 (k-NN) *As previously seen, k-NN has hypothesis stability equal to $\frac{k}{m}$ such that bagging k-NN has stability with respect to classification loss bounded by*

$$2\sum_{j=1}^{m}\frac{j\beta_j}{m}\mathbb{P}_{\mathbf{r}}\left[d(\mathbf{r})=j\right]=2\sum_{j=1}^{m}\frac{j\frac{k}{j}}{m}\mathbb{P}_{\mathbf{r}}\left[d(\mathbf{r})=j\right]=2\frac{k}{m}\sum_{j=1}^{m}\mathbb{P}_{\mathbf{r}}\left[d(\mathbf{r})=j\right]=2\frac{k}{m}$$

So bagging does not improve stability, which is also experimentally verified by Breiman (1996a).

The next proposition establishes the link between the uniform stability of bagging and that of the base learning algorithm for regression. As before, classification can be treated similarly, see (Bousquet and Elisseeff, 2002).

Proposition 4.3 (Random uniform stability of bagging for regression) *Assume that the loss ℓ is B-lipschitzian with respect to its first variable. Let $F_{\mathcal{D},\mathbf{r}}$ be the outcome of a bagging algorithm whose base machine has uniform stability γ_m w.r.t. the ℓ_1 loss function. Then the random uniform stability β_m of $F_{\mathcal{D},\mathbf{r}}$ with respect to ℓ is bounded by*

$$\beta_m \le B\sum_{k=1}^{m}\frac{k\gamma_k}{m}\mathbb{P}_{\mathbf{r}}\left[d(\mathbf{r})=k\right]. \tag{22}$$

Proof The random uniform stability of bagging is given by

$$\beta_m=\sup_{\mathcal{D},z}\left|\mathbf{E}_{\mathbf{r}_1,\ldots,\mathbf{r}_t}\left[\ell\left(\frac{1}{T}\sum_{t=1}^{T}f_{\mathcal{D}(\mathbf{r}_t)},z\right)-\ell\left(\frac{1}{T}\sum_{t=1}^{T}f_{\mathcal{D}^{\setminus i}(\mathbf{r}_t)},z\right)\right]\right|.$$

This can be bound by taking the absolute valued inside the expectation. Then, following the same lines as in the proof of Proposition 4.1 we have:

$$\beta_m \le B\sup_{\mathcal{D},x}\left\{\mathbf{E}_{\mathbf{r}}\left[\Delta(\mathcal{D}(\mathbf{r}),x)\mathbf{1}_{i\in\mathbf{r}}\right]\right\}$$

where, we recall, $\Delta(\mathcal{D}(\mathbf{r}),x)=|f_{\mathcal{D}(\mathbf{r})}-f_{\mathcal{D}^{\setminus i}(\mathbf{r})}|$ and function $\mathbf{1}_{i\in\mathbf{r}}$ is equal to one if point i is sampled during bootstrapping and zero otherwise. We then have

$$\beta_m \le B\,\mathbf{E}_{\mathbf{r}}\left[\sup_{\mathcal{D},x}\left\{\Delta(\mathcal{D}(\mathbf{r}),x)\right\}\mathbf{1}_{i\in\mathbf{r}}\right].$$

Now we observe that

$$\sup_{\mathcal{D},x}\left\{\Delta(\mathcal{D}(\mathbf{r}),x)\right\}=\sup_{\mathcal{D}(\mathbf{r}),x}\left\{\Delta(\mathcal{D}(\mathbf{r}),x)\right\}=\gamma_{d(\mathbf{r})}.$$

Placing this bound in the previous one gives

$$\beta_m \le \mathbf{E}_{\mathbf{r}}\left[\gamma_{d(\mathbf{r})}\mathbf{1}_{i\in\mathbf{r}}\right].$$

The proof is now exactly the same as in the final part of Proposition 4.1. ■

Example 6 (SVM regression) *We have seen in Example 2 that the uniform stability of a SVM w.r.t. the ℓ_1 loss is bounded by $1/\lambda$. The uniform stability of bagging SVM is then roughly bounded by $0.632/\lambda$ if the SVM is trained on all bootstrap sets with the same λ. So that the bound on the random uniform stability of a bagged SVM is better than the bound on the uniform stability for a single SVM trained on the whole training set with the same λ.*

4.2 Subbagging

Subbagging is a variation of bagging where the sets $\mathcal{D}(\mathbf{r}_t)$, $t = 1, \dots, T$ are obtained by sampling $p \leq m$ points from $\mathcal{D}$ *without* replacement. Like in bagging, a base learning algorithm is trained on each set $\mathcal{D}(\mathbf{r}_t)$ and the obtained solutions $f_{\mathcal{D}(\mathbf{r}_t)}$ are combined by average.

The proofs above can then be used here directly which gives the following upper bounds on stability for subbagging:

Proposition 4.4 (Stability of subbagging for regression) *Assume that the loss ℓ is B-lipschitzian w.r.t. its first variable. Let $F_{\mathcal{D},\mathbf{r}}$ be the outcome of a subbagging algorithm whose base machine is symmetric and has uniform (resp. hypothesis or pointwise hypothesis) stability γ_m w.r.t. the ℓ_1 loss function, and subbagging is done by sampling p points without replacement. Then the random uniform (resp. hypothesis or pointwise hypothesis) stability β_m of $F_{\mathcal{D},\mathbf{r}}$ w.r.t. ℓ is bounded by*

$$\beta_m \leq B\gamma_p \frac{p}{m}.$$

For classification, we have also the following proposition, again only for hypothesis or pointwise hypothesis stability as in Section 2:

Proposition 4.5 ((P.) Hypothesis stability of subbagging for classification) *Let $F_{\mathcal{D},\mathbf{r}}$ be the outcome of a subbagging algorithm whose base machine is symmetric and has hypothesis (resp. pointwise hypothesis) stability γ_m with respect to classification loss, and subbagging is done by sampling p points without replacement. Then the random hypothesis (resp. pointwise hypothesis) stability β_m of $F_{\mathcal{D},\mathbf{r}}$ with respect to the ℓ_1 loss function is bounded by*

$$\beta_m \leq 2\gamma_p \frac{p}{m}.$$

4.3 Bounds on the Performance of Subbagging

We can now prove bounds on the performance of bagging and subbagging. We present the following theorems for subbagging but the same statements hold true for bagging where, in the bounds below, $\frac{p\gamma_p}{m}$ is replaced by $\sum_{k=1}^{m} \frac{k\gamma_k}{m} \mathbb{P}_{\mathbf{r}}[d(\mathbf{r}) = k]$ which is roughly equal to $0.632\gamma_{0.632m}$ when m is sufficiently large.

Theorem 16 *Assume that the loss ℓ is B-lipschitzian w.r.t. its first variable. Let $F_{\mathcal{D},\mathbf{r}}$ be the outcome of a subbagging algorithm. Assume subbagging is done with T sets of size p subsampled without replacement from $\mathcal{D}$ and the base learning algorithm has hypothesis stability γ_m and pointwise* *hypothesis stability γ'_m, both stabilities being w.r.t. the ℓ loss. The following bounds hold separately with probability at least $1-\delta$*

$$R_{gen}(F_{\mathcal{D},\mathbf{r}}) \leq R_{\ell oo}(F_{\mathcal{D},\mathbf{r}}) + \sqrt{\delta^{-1}\frac{2M^2+12MBp\gamma_p}{m}} \tag{23}$$

$$R_{gen}(F_{\mathcal{D},\mathbf{r}}) \leq R_{emp}(F_{\mathcal{D},\mathbf{r}}) + \sqrt{\delta^{-1}\frac{2M^2+12MBp\gamma'_p}{m}}. \tag{24}$$

Proof The inequalities follow directly from plugging the result of Proposition 4.4 in Theorems 9 and 12 respectively. ■

Note that, as in Proposition 4.2, the same result holds for classification if we set $B=2$ and $M=1$.

The following theorem holds for regression. The extension to the case of classification can be done again as in (Bousquet and Elisseeff, 2002).

Theorem 17 *Assume that the loss ℓ is B-lipschitzian w.r.t. its first variable. Let $F_{\mathcal{D},\mathbf{r}}$ be the outcome of a subbagging algorithm. Assume subbagging is done with T sets of size p subsampled without replacement from $\mathcal{D}$ and the base learning algorithm has uniform stability γ_m w.r.t. the ℓ loss. The following bounds hold separately with probability at least $1-\delta$ in the case of regression*

$$R_{gen}(F_{\mathcal{D},\mathbf{r}}) \leq R_{\ell oo}(F_{\mathcal{D},\mathbf{r}}) + \frac{Bp\gamma_p}{m} + \left(\frac{M+4B(m/m-1)p\gamma_p}{\sqrt{2m}} + \frac{\sqrt{2}BM}{\sqrt{T}}\right)\sqrt{\log(2/\delta)}, \tag{25}$$

and

$$R_{gen}(F_{\mathcal{D},\mathbf{r}}) \leq R_{emp}(F_{\mathcal{D},\mathbf{r}}) + 2\frac{Bp\gamma_p}{m} + \left(\frac{M+4Bp\gamma_p}{\sqrt{2m}} + \frac{\sqrt{2}BM}{\sqrt{T}}\right)\sqrt{\log 2/\delta}. \tag{26}$$

Proof We recall that $\mathbf{r} = (\mathbf{r}_1,\ldots,\mathbf{r}_T)$ and introducethe notation

$$\mathbf{r}^t = (\mathbf{r}_1,\ldots,\mathbf{r}_{t-1},\mathbf{r}',\mathbf{r}_{t+1},\ldots,\mathbf{r}_T).$$

Note that

$$\left|\ell(F_{\mathcal{D},\mathbf{r}},z) - \ell(F_{\mathcal{D},\mathbf{r}^t},z)\right| = \left|\ell\left(\sum_{s=1}^{T} f_{\mathcal{D}(\mathbf{r}_s)},z\right) - \ell\left(\sum_{s=1,s\neq t}^{T} f_{\mathcal{D}(\mathbf{r}_s)} + f_{\mathcal{D}(\mathbf{r}')},z\right)\right| \leq$$
$$\leq \frac{B}{T}\left|f_{\mathcal{D}(\mathbf{r}')}\right| \leq \frac{B}{T}M$$

Thus, the constant ρ in Theorem 15 is bounded as

$$\rho = \sup_{\mathbf{r},\mathbf{r}'_t}\left|\ell(F_{\mathcal{D},\mathbf{r}},z) - \ell(F_{\mathcal{D},\mathbf{r}^t},z)\right| \leq \frac{B}{T}M.$$

The result then follows by using this theorem and Proposition 4.4. ■

We comment on some characteristics of the above bounds for subbagging:

- In Theorem 16 if, as $m \to \infty$, $\frac{p\gamma_p}{m} \to 0$ then the empirical or leave-one-out error converge to the expected error. In particular, if $p = O(1)$ as $m \to \infty$ the empirical or leave-one-out error converge to the expected one as $O(1/\sqrt{m})$. This convergence is in probability as opposed to the convergence provided by Theorem 17 which is almost surely.
- Although we can derive bounds for bagging using our theory in section 3 that were not possible to derive with the existing theory summarized in Section 2, our results for bagging do not show that bagging actually improves performance. Indeed, for example comparing Theorems 17 and 6, it is not clear which bound is tighter as that depends on the constants (e.g. M, B, and other constants) and the behavior of γ_p as p increases. Developing tighter bounds or lower bounds within our analysis for bagging is needed for this purpose. This is an open problem.
- Theorem 17 indicates that the effects of the number of subsamples T is of the form $\frac{1}{\sqrt{T}}$, so there is no need for a large T, as also observed in practice (Breiman, 1996a). For example, it is sufficient that T scales as $\sqrt{m}$. This result improves upon the analysis of (Evgeniou et al., 2004) where in order to have convergence of the empirical or leave-one-our error to the expected error it was required that T is infinite.
- The bounds provided by Theorem 17 imply that the empirical or leave-one-out error converge to the expected error provided, as $m \to \infty$, that $\frac{p\gamma_p}{\sqrt{m}} \to 0$ *and* $T \to \infty$. The latter condition is not a problem in practice, for example one could choose $T = O(\sqrt{m})$ to get convergence, but it indicates a weak point of the uniform stability analysis as opposed to the hypothesis stability analysis above. As we discussed above, it may be possible to show that parameter ρ appearing in Theorem 15 depends on m for the case of bagging, or to show that this is not possible in which case it will be a limitation of our approach. This is an open problem.

5. Conclusions

We presented a theory of random stability for randomized learning methods that we also applied to study the effects of bagging on the stability of a learning method. This is an extension of the existing theory about the stability and generalization performance of deterministic (symmetric) learning methods (Bousquet and Elisseeff 2002). We note that the setup that we developed for this analysis, such as the issues and assumptions that we considered in Section 3, may be also used for other studies of randomized learning algorithms – such as extensions of other theories about stability from deterministic to randomized learning methods. The bounds we proved show formally the relation of the generalization error to the stability of the (random) algorithm. There is currently no lower bound hence we cannot practically use the bounds when the number of data m is small (e.g., several hundreds or thousands, which is the case in many current applications). This issue concerns both the deterministic (Bousquet and Elisseeff, 2002) as well as the random case. Developing tighter bounds as well as lower bounds in order to be able to use the theory developed here in practice is an open question.

Acknowledgments

We wish to thank Sayan Mukherjee, Tomaso Poggio, Ryan Rifkin, and the anonymous reviewers for their useful comments.

Appendix A. Proofs of Lemmas 3.1 and 3.2

The proofs of Lemmas 3.1 and 3.2 follow directly the proof that has been given in (Bousquet and Elisseeff, 2002). We reproduce the proof here with the changes that are required to handle non symmetric algorithms. Before entering the core of the calculations, let us introduce some convenient notation. We will denote by

$$\ell_{ij}(z,z',z'') = \ell(f_{\mathcal{D}_{ij}(z,z')},z'') \tag{27}$$

the loss of an algorithm A trained on

$$\mathcal{D}_{i,j}(z,z') = \left(z_1,\ldots,z_{i-1},z,z_{i+1},\ldots,z_{j-1},z',z_{j+1},\ldots,z_m\right)$$

which represents the training set $\mathcal{D}$ where z_i and z_j have been replaced by z and z'. When $i = j$, it is required that $z = z'$. Note that the position of z_i and z_j matters here since the algorithm is not symmetric. Since we have $\mathcal{D}_{i,j}(z_i,z_j) = \mathcal{D}_{k,l}(z_k,z_l)$ for any i, j and k,l in $\{1,\ldots,m\}$, we use the notation $\ell(z)$ to denote $\ell_{ij}(z_i,z_j,z)$ for all i and j in $\{1,\ldots,m\}$. According to these notations we have

$$\ell_{ij}(\emptyset,z_j,z_i) = \ell(f_{\mathcal{D}^{\setminus i}},z_i),$$

that is, we replace z_i by the empty set when it is removed from the training set. Since $\ell_{ij}(\emptyset,z_j,z_i)$ does not depend on j, we will denote it by ℓ_i.

Different tricks such as decomposing sums, renaming and permuting variables will be used in the following calculations. Since the proofs are very technical and mostly formal, we explain here more precisely what these steps are. Decomposing sums is the main step of the calculations. The idea is to transform a difference $a-b$ into a sum $a-b = \sum_{i=1}^{k} a_i - a_{i+1}$ ($a_1 = a$ and $a_{k+1} = b$) so that the quantities $a_i - a_{i+1}$ in the sum can be bounded by terms of the form $\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z,z_j,z_i)-\ell(z_i)\right|\right]$, the latter being directly related to the notion of stability we defined. Renaming variables corresponds to simply changing the name of one variable into another one. Most of time, this change will be done between z, z_i and z_j using the fact that z and the z_i's are independently and identically distributed so that averaging w.r.t. z is the same as w.r.t. z_i. The last technique we use is symmetrization. The following simple lemma will allow us to perform some symmetrization without changing significantly the outcome of a (stable) learning algorithm.

Lemma 18 *Let A be a (non-symmetric) algorithm and let ℓ be as defined in Equation (27), we have* $\forall (i,j) \in \{1,\ldots,m\}^2$

$$\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(z)-\ell_{ij}(z_j,z_i,z)\right|\right] \le \frac{3}{2}\left(\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell(z)\right|\right] + \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z_i,z',z)-\ell(z)\right|\right]\right). \tag{28}$$

Proof We have

$$\begin{aligned}\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(z)-\ell_{ij}(z_j,z_i,z)\right|\right] \le{}& \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell(z)-\ell_{ij}(z',z_j,z)\right|\right]\\ &+\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell_{ij}(z',z_i,z)\right|\right]+\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_i,z)-\ell_{ij}(z_j,z_i,z)\right|\right]\end{aligned} \tag{29}$$

Since the distribution over $\mathcal{D}$ is i.i.d., integrating with respect to z_i is the same as integrating w.r.t. z_j or z', and we can swap the role of z' and z_i in the second term of the r.h.s. , and of z_i and z_j in the last term.

$$\begin{aligned}\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell_{ij}(z',z_i,z)\right|\right] &= \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell(z)-\ell_{ij}(z_i,z',z)\right|\right]\\ \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_i,z)-\ell_{ij}(z_j,z_i,z)\right|\right] &= \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell(z)\right|\right],\end{aligned}$$

which gives the following result:

$$\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(z)-\ell_{ij}(z_j,z_i,z)\right|\right] \quad \leq \quad 2\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z',z_j,z)-\ell(z)\right|\right] \;+\; \mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,z',z)-\ell(z)\right|\right] \quad (30)$$

If instead of (29) we used the following decomposition,

$$\begin{aligned}\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(z)-\ell_{ij}(z_j,z_i,z)\right|\right] \leq \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell(z)-\ell_{ij}(z_i,z',z)\right|\right] \\ +\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z_i,z',z)-\ell(z_j,z',z)\right|\right]+\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell(z_j,z',z)-\ell_{ij}(z_j,z_i,z)\right|\right],\end{aligned}$$

it would have led to

$$\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(z)-\ell_{ij}(z_j,z_i,z)\right|\right] \leq \mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z',z_j,z)-\ell(z)\right|\right]+2\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,z',z)-\ell(z)\right|\right].$$

Averaging this inequality with (30), we get the final result. ■

Note that the quantity appearing in the r.h.s. of Equation (28) can be bounded by different quantities related to pointwise hypothesis stability or to hypothesis stability. We have indeed

$$\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(z)-\ell_{ij}(z_j,z_i,z)\right|\right] \leq 3\left(\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z,z_j,z_i)-\ell(z_i)\right|\right] \; +\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,z,z_j)-\ell(z_j)\right|\right]\right), \quad (31)$$

which is related to the definition of pointwise hypothesis stability and will be used when the focus is on empirical error. We have also

$$\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(z)-\ell_{ij}(z_j,z_i,z)\right|\right] \leq 3\left(\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(\emptyset,z_j,z)-\ell(z)\right|\right] \; +\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,\emptyset,z)-\ell(z)\right|\right]\right),$$

which is related to bounds on the leave-one-out error. Both bounds have the same structure and it will turn out that the following calculations are almost identical for leave-one-out error and empirical error. We can now start the main part of the proofs. The notations are difficult to digest but the ideas are simple and use only the few formal steps we have described before. We first state the following lemma as in (Bousquet and Elisseeff, 2002):

Lemma 19 *For any (non-symmetric) learning algorithm A, we have*

$$\begin{aligned}\mathbf{E}_{\mathcal{D}}\left[(R_{gen}-R_{emp})^2\right] \leq \frac{1}{m^2}\sum_{i\neq j}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-\frac{2}{m^2}\sum_{i\neq j}^{m}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right] \\ +\frac{1}{m^2}\sum_{i\neq j}\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)\ell(z_j)\right]+\frac{1}{m^2}\sum_{i=1}^{m}\left(\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-2\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]+\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)^2\right]\right)\end{aligned}$$

and

$$\begin{aligned}\mathbf{E}_{\mathcal{D}}\left[(R_{gen}-R_{\ell oo})^2\right] \leq \frac{1}{m^2}\sum_{i\neq j}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-\frac{2}{m^2}\sum_{i\neq j}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell_i\right] \\ +\frac{1}{m^2}\sum_{i\neq j}\mathbf{E}_{\mathcal{D}}\left[\ell_i\ell_{ij}(z_i,\emptyset,z_j)\right] \\ +\frac{1}{m^2}\sum_{i=1}^{m}\left(\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-2\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell_i\right]+\mathbf{E}_{\mathcal{D}}\left[\ell_i^2\right]\right).\end{aligned}$$

Proof We have

$$\begin{aligned}
\mathbf{E}_{\mathcal{D}}\left[R_{gen}^2\right] &= \mathbf{E}_{\mathcal{D}}\left[\mathbf{E}_z \ell(z)^2\right] \\
&= \mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right] \\
&= \frac{1}{m^2}\sum_{i \neq j}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right] + \frac{1}{m^2}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right],
\end{aligned}$$

and also

$$\begin{aligned}
\mathbf{E}_{\mathcal{D}}\left[R_{gen}R_{emp}\right] &= \mathbf{E}_{\mathcal{D}}\left[R_{gen}\frac{1}{m}\sum_{i=1}^{m}\ell(z_i)\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D}}\left[R_{gen}\ell(z_i)\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right] \\
&= \frac{1}{m^2}\sum_{i \neq j}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right] + \frac{1}{m^2}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right],
\end{aligned}$$

and also

$$\begin{aligned}
\mathbf{E}_{\mathcal{D}}\left[R_{gen}R_{\ell oo}\right] &= \mathbf{E}_{\mathcal{D}}\left[R_{gen}\frac{1}{m}\sum_{i=1}^{m}\ell_i\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D}}\left[R_{gen}\ell_i\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell_i\right] \\
&= \frac{1}{m^2}\sum_{i \neq j}^{m}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell_i\right] + \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell_i\right].
\end{aligned}$$

Also we have

$$\mathbf{E}_{\mathcal{D}}\left[R_{emp}^2\right] = \frac{1}{m^2}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)^2\right] + \frac{1}{m^2}\sum_{i \neq j}\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)\ell(z_j)\right]$$

and

$$\mathbf{E}_{\mathcal{D}}\left[R_{\ell oo}^2\right] = \frac{1}{m^2}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D}}\left[\ell_i^2\right] + \frac{1}{m^2}\sum_{i \neq j}\mathbf{E}_{\mathcal{D}}\left[\ell_i\ell_{ij}(z_i, \emptyset, z_j)\right],$$

which concludes the proof. ■

Continuing the proof of Lemma 3.2, we now formulate the first inequality of Lemma 19 as

$$\begin{aligned}\mathbf{E}_{\mathcal{D}}\left[(R_{gen}-R_{emp})^2\right] \leq \frac{1}{m^2}\sum_{i\neq j}\underbrace{\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]}_{I}\\ +\frac{1}{m^2}\sum_{i\neq j}^{m}\underbrace{\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)\ell(z_j)\right]-\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]}_{J}\\ +\frac{1}{m^2}\sum_{i=1}^{m}\underbrace{\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-2\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]+\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)^2\right]}_{K}.\end{aligned}$$

Using the fact that the loss is bounded by M, we have

$$\begin{aligned}K &= \mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\left(\ell(z')-\ell(z_i)\right)\right]+\mathbf{E}_{\mathcal{D},z}\left[\ell(z_i)\left(\ell(z_i)-\ell(z)\right)\right]\\ &\leq 2M^2.\end{aligned}$$

Now we rewrite I as

$$\begin{aligned}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]=\\ =\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')-\ell_{ij}(z',z_j,z)\ell_{ij}(z',z_j,z')\right],\end{aligned}$$

where we renamed z_i as z' in the second term. We have then

$$\begin{aligned}I=\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell(z)-\ell_{ij}(z,z_j,z))\ell(z')\right]\\ +\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z_j,z)-\ell_{ij}(z',z_j,z))\ell(z')\right]\\ +\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell(z')-\ell_{ij}(z',z_j,z'))\ell_{ij}(z',z_j,z)\right].\end{aligned}$$

Thus,

$$|I|\leq 3M\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z,z_j,z)-\ell(z)\right|\right]. \qquad (32)$$

Next we rewrite J as

$$\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)\ell(z_j)\right]-\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]=\mathbf{E}_{\mathcal{D},z,z'}\left[\ell_{ij}(z,z',z)\ell_{ij}(z,z',z')-\ell(z)\ell(z_i)\right]$$

where we renamed z_j as z' and z_i as z in the first term. We have also

$$J=\mathbf{E}_{\mathcal{D},z,z'}\left[\ell_{ij}(z,z',z)\ell_{ij}(z,z',z')-\ell_{ij}(z',z_i,z)\ell_{ij}(z',z_i,z')\right]$$

where we renamed z_i as z' and z_j as z_i in the second term. Using Equation 31, we have

$$\begin{aligned}J\leq\underbrace{\mathbf{E}_{\mathcal{D},z,z'}\left[\ell_{ij}(z,z',z)\ell_{ij}(z,z',z')-\ell_{ij}(z_i,z',z)\ell_{ij}(z',z_i,z')\right]}_{J_1}\\ +3M\left(\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z,z_j,z_i)-\ell(z_i)\right|\right]+\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,z,z_j)-\ell(z_j)\right|\right]\right). \qquad (33)\end{aligned}$$

Let us focus on J_1, we have

$$\begin{aligned}J_1=\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z',z')-\ell_{ij}(z,z_i,z')\ell_{ij}(z,z',z)\right]\\ +\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z',z)-\ell_{ij}(z_i,z',z))\ell_{ij}(z,z_i,z')\right]\\ +\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z_i,z')-\ell_{ij}(z',z_i,z'))\ell_{ij}(z_i,z',z)\right]\end{aligned}$$

and

$$J_1 = \mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z_i,z_j,z_j) - \ell_{ij}(z_i,z,z_j))\ell_{ij}(z_i,z_j,z_i)\right] \\ + \mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z_i,z_j,z_i) - \ell_{ij}(z,z_j,z_i))\ell_{ij}(z_i,z,z_j)\right] \\ + \mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z_j,z_i) - \ell_{ij}(z_i,z_j,z_i))\ell_{ij}(z_j,z_i,z)\right]$$

where we replaced z by z_i, z_i by z and z' by z_j in the first term, and z by z_i and z' by z_j and z_i by z in the second term and, in the last term, we renamed z' by z_i and z_i by z_j. Thus,

$$|J_1| \leq 2M\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z,z_j,z_i) - \ell(z_i)\right|\right] + M\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z_i,z,z_j) - \ell(z_j)\right|\right]. \tag{34}$$

Summing Equation (32) with the inequality on J derived from Equations (34) and (33), we obtain

$$I + J \quad \leq \quad 8M\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z,z_j,z_i) - \ell(z_i)\right|\right] + 4M\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,z,z_j) - \ell(z_j)\right|\right].$$

To bound $I+J$, we can swap the role of i and j (note that i and j are under a sum and that we can permute the role of i and j in this sum without changing anything). In that case, we obtain

$$I + J \leq 4M\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z,z_j,z_i) - \ell(z_i)\right|\right] + 8M\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,z,z_j) - \ell(z_j)\right|\right].$$

Averaging over this bound and the previous one, we finally obtain

$$I + J \leq 6M\left(\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z,z_j,z_i) - \ell(z_i)\right|\right] + \mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,z,z_j) - \ell(z_j)\right|\right]\right).$$

The above concludes the proof of the bound for the empirical error (Lemma 3.2).

The bound for the leave-one-out error (Lemma 3.1) can be obtained in a similar way. Indeed, 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 $\ell_{ij}(z,z',z)$ by $\ell_{ij}(\emptyset,z',z)$. 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 3.4

Proof We first prove Equation (17) and then show how to derive Equation (19). Both proofs are very similar except for some calculations.

Let $K(\mathcal{D},\mathbf{r}) = R_{gen}(f_{\mathcal{D},\mathbf{r}}) - R_{emp}(f_{\mathcal{D},\mathbf{r}})$ the random variable which we would like to bound. For this purpose, we first show that K is close to its expectation w.r.t. $\mathbf{r}$ and then show how this average algorithm is controlled by its stability.

For every $\mathbf{r},\mathbf{s} \in \mathcal{R}^T$, and $T \in \mathbb{N}$, we have

$$|K(\mathcal{D},\mathbf{r}) - K(\mathcal{D},\mathbf{s})| =$$

$$= \left|\mathbf{E}_z\left[\ell(f_{\mathcal{D},\mathbf{r}},z) - \ell(f_{\mathcal{D},\mathbf{s}},z)\right] - \frac{1}{m}\sum_{i=1}^m \left(\ell(f_{\mathcal{D},\mathbf{r}},z_i) - \ell(f_{\mathcal{D},\mathbf{s}},z_i)\right)\right|$$

$$\leq \mathbf{E}_z\left[\left|\ell(f_{\mathcal{D},\mathbf{r}},z) - \ell(f_{\mathcal{D},\mathbf{s}},z)\right|\right] + \frac{1}{m}\sum_{i=1}^m \left|\ell(f_{\mathcal{D},\mathbf{r}},z_i) - \ell(f_{\mathcal{D},\mathbf{s}},z_i)\right|.$$

Thus, using the definition of ρ, this equation implies (when $\mathbf{r}$ and $\mathbf{s}$ differ only in one of the T coordinates) that

$$\sup_{\mathbf{r}_1,\ldots,\mathbf{r}_T,\mathbf{r}'_t} \left|K(\mathcal{D},\mathbf{r}_1,\ldots,\mathbf{r}_T) - K(\mathcal{D},\mathbf{r}_1,\ldots,\mathbf{r}_{t-1},\mathbf{r}'_t,\mathbf{r}_{t+1},\ldots,\mathbf{r}_T)\right| \leq 2\rho$$

and applying Theorem 14 we obtain (note that $\mathcal{D}$ is independent of $\mathbf{r}$)

$$\mathbb{P}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r}) - \mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] \geq \varepsilon \mid \mathcal{D}\right] \leq \exp\left\{-\varepsilon^2/2T\rho^2\right\}.$$

We also have

$$\mathbf{E}_{\mathcal{D}}\left[\mathbb{P}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r}) - \mathbf{E}_{\mathbf{r}}K(\mathcal{D},\mathbf{r}) \geq \varepsilon\right]\right] =$$
$$= \mathbf{E}_{\mathcal{D}}\left[\mathbb{P}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r}) - \mathbf{E}_{\mathbf{r}}K(\mathcal{D},\mathbf{r}) \geq \varepsilon \mid \mathcal{D}\right]\right] \leq \exp\left\{-\varepsilon^2/2T\rho^2\right\}.$$

Setting the r.h.s. equal to δ and writing ε as a function of δ we have that with probability at least $1-\delta$ w.r.t. the random sampling of $\mathcal{D}$ and $\mathbf{r}$:

$$K(\mathcal{D},\mathbf{r}) - \mathbf{E}_{\mathbf{r}}K(\mathcal{D},\mathbf{r}) \leq \sqrt{2T}\rho\sqrt{\log(1/\delta)}. \tag{35}$$

We first bound the expectation of $K(\mathcal{D},\mathbf{r})$. We define $G(\mathcal{D},z) := \mathbf{E}_{\mathbf{r}}[\ell(f_{\mathcal{D},\mathbf{r}},z)]$. We have

$$\begin{aligned}
\mathbf{E}_{\mathcal{D},\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] &= \mathbf{E}_{\mathcal{D}}\left[\mathbf{E}_z\left[G(\mathcal{D},z) - \frac{1}{m}\sum_{i=1}^m G(\mathcal{D},z_i)\right]\right] \\
&= \mathbf{E}_{\mathcal{D},z}\left[G(\mathcal{D},z)\right] - \frac{1}{m}\sum_{i=1}^m \mathbf{E}_{\mathcal{D}}\left[G(\mathcal{D},z_i)\right] \\
&\overset{(a)}{\leq} 2\beta_m + \mathbf{E}_{\mathcal{D}^{\setminus i},z}\left[G(\mathcal{D}^{\setminus i},z)\right] - \frac{1}{m}\sum_{i=1}^m \mathbf{E}_{\mathcal{D}}\left[G(\mathcal{D}^{\setminus i},z_i)\right] \\
&\overset{(b)}{=} 2\beta_m
\end{aligned} \tag{36}$$

where (a) is derived from the fact that the algorithm has random uniform stability β_m, that is,

$$\sup_{\mathcal{D},z}\left|G(\mathcal{D},z) - G(\mathcal{D}^{\setminus i},z)\right| \leq \beta_m,$$

and (b) comes from $\mathbf{E}_{\mathcal{D}}\left[G(\mathcal{D}^{\setminus i},z_i)\right] = \mathbf{E}_{\mathcal{D}^{\setminus i},z}\left[G(\mathcal{D}^{\setminus i},z)\right]$ (it amounts to changing z_i into z). We would like now to apply Theorem 14 to $\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right]$. To this aim, we bound (recall that $\mathcal{D}^i = \mathcal{D}^{\setminus i} \cup z'$):

$$\begin{aligned}
&\left|\mathbf{E}_{\mathbf{r}}\left[\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r}) - K(\mathcal{D}^i,\mathbf{r})\right]\right]\right| = \\
&\quad\Bigg|\underbrace{\frac{1}{m}\left(\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^i,\mathbf{r}},z')\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D},\mathbf{r}},z_i)\right]\right)}_{(a)} + \frac{1}{m}\sum_{i\neq j}\underbrace{\mathbf{E}_{\mathbf{r}}[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z_j)] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D},\mathbf{r}},z_j)\right]}_{(b)} \\
&\quad + \frac{1}{m}\sum_{i\neq j}\underbrace{\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^i,\mathbf{r}},z_j)\right] - \mathbf{E}_{\mathbf{r}}[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z_j)]}_{(c)} + \underbrace{\mathbf{E}_{\mathbf{r}}\left[\mathbf{E}_z\left[\ell(f_{\mathcal{D},\mathbf{r}},z) - \ell(f_{\mathcal{D}^i,\mathbf{r}},z)\right]\right]}_{(d)}\Bigg|
\end{aligned} \tag{37}$$

where (a) is bounded by $\frac{M}{m}$, (b), (c) are bounded by β_m and (d) is similarly bounded by $2\beta_m$. So that $\sup_{\mathcal{D},z',z} \left|\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] - \mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D}^i,\mathbf{r})\right]\right| \leq \frac{M}{m} + 4\beta_m$ and we derive that

$$\mathbb{P}_{\mathcal{D}}\left[\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] \geq \varepsilon + 2\beta_m\right] \leq \exp\left\{-\frac{2m\varepsilon^2}{(M+4m\beta_m)^2}\right\},$$

which implies that with probability at least $1-\delta$ w.r.t. the random sampling of $\mathcal{D}$ and r

$$\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] \leq 2\beta_m + \frac{M+4m\beta_m}{\sqrt{2m}}\sqrt{\log(1/\delta)}. \tag{38}$$

Observe that the inequalities in Equations (35) and (38) hold simultaneously with probability at least $1-2\delta$. The result follows by combining those inequalities and setting $\delta = \delta/2$.

The proof of Equation (19) follows the same reasoning except that the chain of Equations (36) and (37) are different. We have

$$\begin{aligned}
\mathbf{E}_{\mathcal{D},\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] &= \mathbf{E}_{\mathcal{D}}\left[\mathbf{E}_z\left[G(\mathcal{D},z)\right] - \frac{1}{m}\sum_{i=1}^{m} G(\mathcal{D}^{\setminus i}, z_i)\right] \\
&= \mathbf{E}_{\mathcal{D},z}\left[G(\mathcal{D},z)\right] - \frac{1}{m}\sum_{i=1}^{m} \mathbf{E}_{\mathcal{D},z}\left[G(\mathcal{D}^{\setminus i}, z)\right] \\
&\leq \beta_m,
\end{aligned}$$

and denoting $\mathcal{D}^{\setminus i,j}$ the set $\mathcal{D}$ where z_i and z_j have been removed, and $\mathcal{D}^{i\setminus j}$ the set $\mathcal{D}^i$ where z_j has been removed (for $j \neq i$),

$$\begin{aligned}
&\left|\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] - \mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D}^i,\mathbf{r})\right]\right| = \\
&\quad\left|\underbrace{\frac{1}{m}\left(\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}}, z_i)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}}, z')\right]\right)}_{(a)} + \frac{1}{m}\sum_{i\neq j}\underbrace{\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus j},\mathbf{r}}, z_j)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i,j},\mathbf{r}}, z_j)\right]}_{(b)}\right. \\
&\qquad\left. + \frac{1}{m}\sum_{j\neq j}^{m}\underbrace{\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i,j},\mathbf{r}}, z_j)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{i\setminus j},\mathbf{r}}, z_j)\right]}_{(c)} + \underbrace{\mathbf{E}_{\mathbf{r}}\left[\mathbf{E}_z\left[\ell(f_{\mathcal{D},\mathbf{r}}, z) - \ell(f_{\mathcal{D}^i,\mathbf{r}}, z)\right]\right]}_{(d)}\right|.
\end{aligned}$$

Finally, note that (a) is bounded by $\frac{M}{m}$, (b) and (c) are bounded by β_{m-1} and (d) by $2\beta_m$. ■

References

[1] Andonova, S., Elisseeff, A., Evgeniou, T., and Pontil, M. (2002), "A simple algorithm to learn stable machines", Proceedings of the 15th European Conference on Artificial Intelligence (ECAI) 2002.

[2] Bousquet, O., and Elisseeff, A. (2002), "Stability and generalization", *Journal of Machine Learning Research*, 2:499–526.

[3] Breiman, L. (1996a), “Bagging predictors”, *Machine Learning*, 26(2):123–140.

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

[5] Devroye, L., Györfi, L., and Lugosi, G. (1996), *A Probabilistic Theory of Pattern Recognition*, Number 31 in Applications of Mathematics, Springer, New York.

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

[7] Evgeniou, T., Pontil, M., and Elisseeff, A. (2004), “Leave-one-out error, stability, and generalization of voting combinations of classifiers”, *Machine Learning*, 55:1, 2004 .

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

[9] Kutin, S., and Niyogi, P. (2002), “Almost-everywhere algorithmic stability and generalization error”, *Uncertainty in Artificial Intelligence (UAI)*, August, 2002, Edmonton, Canada.

[10] McDiarmid, C. (1989), “On the method of bounded differences”, In *Survey in Combinatorics*, p. 148–188. Cambridge University Press, Cambridge.

[11] Poggio, T., and Girosi, F. (1990), “Networks for approximation and learning”, *Proceedings of the IEEE*, 78 (9).

[12] Poggio, T., Rifkin, R., Mukherjee, S. and Niyogi, P. (2004), “Learning Theory: general conditions for predictivity”, *Nature*, Vol. 428, 419-422.

[13] Vapnik, V. (1998), *Statistical Learning Theory*. Wiley, New York, 1998.