Stability of Randomized Learning Algorithms with an Application to Bootstrap Methods

Andre Elisseeff
Max Planck Institute for Biological Cybernetics, Tuebingen, Germany
andre.elisseeff@tuebingen.mpg.de

Theodoros Evgeniou
INSEAD, Fontainebleau, France,
theodoros.evgeniou@insead.edu

Massimiliano Pontil
Department of Computer Sciences, University College London
Gower Street, London WC1E, UK
m.pontil@cs.ucl.ac.uk

Abstract

The purpose of this paper is twofold: first to study the predictive performance of randomized learning methods using notions of stability, namely how much changes in the training data influence the estimated models; second, to use these general results in order to study bootstrap methods. We first give formal definitions of stability for randomized methods, and we 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 methods that depend on their random stability. We then use these general results to study the effects of bagging (**B**ootstrap **Agg**regat**ing**) on the stability of a learning method and to give non-asymptotic bounds on the predictive performance of bagging. We consider bagging in the case where the base machines treat multiples copies of a training point as one point. We also show that if bagging is done using small sub-samples of the original data, we call this subagging, the effects on stability are larger and the bounds on the difference between empirical and expected error are tighter.

1 Introduction

One of the key motivations for this paper is to develop a theoretical analysis of bootstrap methods, such as bagging. These methods are randomized, so we first develop a general theory for randomized learning methods. We then apply this theory to the study of combinations of methods in the spirit of bagging.

Combining estimation methods instead of using a single one is an idea used by many researchers in recent years [5, 19, 20, 37, 27]. Many techniques have been designed among which the Arcing (**A**daptive **R**esampling **C**ombin**ing**) methods [7]. The latter consist in taking the sign of a weighted average of different base machines. Depending on the way the weights are computed and the machines are trained, we can get two standard ensemble methods, namely bagging [5] or boosting [19]. There has been many other practical efforts to understand the effect of bagging and particularly the difference between bagging and boosting [30, 7, 2, 18].

Bagging is based on the idea of bootstrapping [14]. Breiman [5] used bagging to improve the generalization performance of tree classifiers. The idea is to average (since for classification, voting and averaging the $\{-1,+1\}$ output of a classifier are the same, we use both terms indifferently) uniformly the $\{-1,+1\}$ outputs of many classifiers that have been each trained on a data set built from a random draw of m samples with replacement from a training set of size m. One motivation for such a method has been given by Breiman by invoking bias and variance arguments: bagging reduces the variance and does not increase too much the bias [5]. Hence, it reduces the sum of both which appears in a decomposition of the generalization error or is directly related to it. The bias/variance reasoning has been applied in different contexts [5, 7, 25] and has been discussed by Freund and Schapire [18] who pointed out some of the weaknesses of the approach.

It is important to mention that the definitions of variance vary from one author to another. Breiman in [7] gives a definition that suits the classification context but which is not clearly related to the classical definition of the variance of a random variable. In regression and for quadratic loss, Breiman [5] shows the link between the generalization error and a variance which this time is the "classical" variance. Since there are many concepts named as variance, we will keep from now on the classical meaning of this word, as it has been used by Friedman and Hall [21] and by Bühlmann and Yu [9]. These authors developed an asymptotic analysis of bagging. Focusing on particular examples such as MARS or Decision Trees, Bühlmann and Yu compute the asymptotic variance and show that it is reduced when bagging is used.

Along with the practical considerations, the intuition given by Breiman in [5] gives an interesting insight: the effect of bagging depends on the "stability" of the base classifier. Stability means here changes in the output of the classifier when the training set is perturbed. If the base classifiers are stable, then bagging is not expected to decrease the generalization error. On the other hand, if the base classifier is unstable such as decision trees, the generalization performance is supposed to be increased with bagging.

The notion of stability that has been heuristically defined by Breiman is very close to other stability definitions that have been used by different researchers but for other purposes. A simple special case of stability has be used in [24] to study the bias and variance of cross validation and bootstrap error estimates [13, 15], without however giving any non-asymptotic bounds on these errors in the general case. Devroye and Wagner [12] derived non-asymptotic bounds on the generalization error of k-Nearest Neighbor (k-NN) in terms of stability and showed at the same time, that k-NN as well as other local rules are stable. Their work as well as other stability related results have been summarized in [11]. More recently, Kearns and Ron [23] used a refined notion of stability to derive sanity check bounds: the bounds with stability are not worse than those you can obtain from the theory of Vapnik and Chervonenkis [35, 36]. They proved that a classifier chosen from a hypothesis space with finite VC-dimension is stable to some extent. Bousquet and Elisseeff [4] related stability and generalization performance of many algorithms by showing that regularized learning methods are generally stable. In this paper we first extend the results of [4] to the case of randomized learning methods, and then we apply them to the case of bagging. An analysis of bagging using deterministic notions of stability was done in [17]. In that work an asymptotic analysis of bagging where an infinite number of methods is combined was presented. Here we discuss the case of finite combinations using random stability notions.

1.1 Outline of the paper

Bagging as used in practice is a randomized algorithm which requires a particular setup. To apply the same approach as in [4] which is described in section 2, we need first to extend it to randomized algorithms (section 3). Then we prove that stability notions for randomized algorithms can be used to control the deviation between empirical or leave-one-out error and true error.

We then apply this stability formal setup to the analysis of bagging in section 4, where we compute the stability of bagging and subagging methods. We thus show a formal relationship between bagging and stability as it has been previously sketched by Breiman: bagging can increase the stability of the learning machines when these are not stable. We don't present

any evidence in favor of the generalization improvement of bagging. The main fact we argue is that for unstable methods bagging decreases the difference between empirical (or leave-one-out) and test error.

We also study a variation of bagging where instead of having each machine using a sub-sample with replacement of size equal to that of the original training set, we let each machine use only a small part (i.e. 10-50%) of the original training data formed via random sampling without replacement. This variant has already been suggested in [9, 21, 32] and is called *subagging* (**Sub**sample **Agg**regat**ing**). We formally show that for subagging the bounds we get are tighter than for standard bagging, and depend on the ratio of the size of the sub-samples to the size of the total training set.

It is interesting to note that the *non*-asymptotic results presented here are in agreement with the asymptotic results on bootstrap error estimates using small sub-samples discussed in [32]. Although we do not study this here, we believe that our results for bagging and subagging using the random hypothesis stability can be extended in order to study the relation of the non-asymptotic accuracy of various bootstrap error estimates and the stability of a learning method in general. This requires first an extension of our results to the case where multiple points of bootstrap samples are treated as such - and not as single points. We leave these as open questions.

1.2 Basic notation

In the following, calligraphic font is used for sets and capital letters refer to numbers unless explicitly defined. Let $\mathcal{X}$ and $\mathcal{Y}$ be two Hilbert spaces 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, for example linear least square estimation, 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. 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), and we denote $\mathcal{D}^i$ the training set obtained by changing point (x_i, y_i) from $\mathcal{D}$ into $z' := (x', y')$, that is the set $(\mathcal{D} \setminus z_i) \cup z'$. $f_{\mathcal{D}}$ is sometimes denoted by f and $f_{\mathcal{D}^i}$ by f_i.

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

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

We define as well the *empirical error*:

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

Finally we define the *leave-one-out error* as:

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

Note that these errors are all 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, e.g. we assume that there exist $A > 0$ such that for every $y, y_1, y_2 \in \mathcal{Y}$ we have $|\ell(y_1, y) - \ell(y_2, y)| \leq A|y_1 - y_2|$. Note that the absolute value satisfies this condition with $A = 1$, whereas the square loss satisfies the condition if $\mathcal{Y}$ is compact.

2 Stability and generalization for deterministic algorithms

As we mentioned in the introduction, stability has been used in machine learning and statistics since the late seventies. Basically, all the definitions we will give follow the same intuition as what Breiman defined in his papers [6, 7]: stability measures how the output of a learning system changes when the training set is perturbed, i.e. when one element in the training set is removed or replaced. In this section we briefly review the results in [12, 23, 4] that show that stability is linked to generalization. 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 Hypothesis stability

The first notion of stability we consider has been stated in [4] and is inspired by the work of Devroye and Wagner [12]. It is very close to what Kearns and Ron [23] defined as hypothesis stability:

Definition 2.1 (Hypothesis Stability) *An algorithm A has* hypothesis stability β_m *w.r.t. the loss function ℓ if the following holds:*

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

It can be shown [11, 4] that many algorithms are stable according to this definition. Here is an example.

Example 2.1 (Hypothesis Stability of k-NN) *With respect to the classification loss, k-NN is $\frac{k}{m}$ stable. This can be seen via symmetrization arguments. For the sake of simplicity we give here the proof for the 1-NN. 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 nearest neighbor machine 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 $\mathbf{1}_A$ is the indicator function of set A. 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 have then:

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

Note that v_i depends on $\mathcal{D}$. Now averaging over $\mathcal{D}$ we need to compute $\mathbf{E}_{\mathcal{D}}\left[\mathbb{P}(v_i)\right]$ 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 have then:

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

So that: $\mathbf{E}_{\mathcal{D}}\left[\mathbb{P}(v_i)\right] = \frac{1}{m}$. And finally, the 1-NN has a hypothesis stability bounded above by $1/m$.

In the following, we will say that an algorithm is stable when its stability scales like $\frac{1}{m}$. It can be shown [4] that when an algorithm has a 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 leave-one-out error and the expected error holds:

Theorem 2.1 (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 ℓ such that $0 \leq \ell(f, z) \leq M$). Then with probability $1 - \eta$*

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

The proof consists of first bounding the second order momentum of $(R_{gen}[f_{\mathcal{D}}] - R_{\ell oo}[f_{\mathcal{D}}])$ and then applying Chebyshev's inequality. So a similar bound on $(R_{gen}[f_{\mathcal{D}}] - R_{\ell oo}[f_{\mathcal{D}}])^2$ holds, as we plot in Figure 1 below.

Theorem 2.1 holds for any loss functions as soon as stability can be proved w.r.t. this loss function. For k-NN used for classification, we have then with probability $1 - \eta$:

$$R_{gen}[f_{\mathcal{D}}] \leq R_{\ell oo}[f_{\mathcal{D}}] + \sqrt{\eta^{-1}\frac{6k + 1}{2m}} \tag{2}$$

According to this bound, for two identical leave-one-out errors, smaller generalization error should be related to smaller k. For k-NN, the leave-one-out error seems to be more informative when the classifier is computed more "locally" (that is with a small k), so that leaving one point out does not change too much the decision function.

Figure 1: *Second order momentum of the difference between leave-one-out error and generalization errors for k-NN ($k = 1, .., 5$) on the ionosphere data set. The bars represent the 95% confidence interval for each k of a two sided χ^2 test.*

To illustrate this point further, we plot in figure 1 the squared difference between leave-one-out error and generalization error for k-NN, $k = 1, .., 5$. The errors are computed on the UCI ionosphere data set. They are averaged over 500 runs, each run consisting in splitting randomly the data set into one training set of size 35 and one test set of size 315. The training set is used to train the k-NN and to compute the leave-one-out errors. This experimental setting is equivalent to considering that $\mathcal{Z}$ is the finite ionosphere data set and $\mathbb{P}$ corresponds to a uniform draw without replacement of m points. For this particular setting, we observe that $(R - R_{loo})^2$ increases for larger k which is consistent with what the stability results say.

Similar bounds can be derived for the empirical error when a slightly different notion of stability is used [4].

Definition 2.2 (Pointwise hypothesis stability) *An algorithm A has* pointwise hypothesis stability β_m *w.r.t. the loss function ℓ if the following holds:*[1]

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

Note that the 1-NN does not have a "good" pointwise hypothesis stability since $\ell(f_{\mathcal{D}}, z_i)$ is always equal to zero. With respect to the classification loss, the pointwise hypothesis stability corresponds then to the expectation of the leave one out error which may not be computed a priori.

As for the case of hypothesis stability and leave-one-out error above, it can also be shown [4] that when an algorithm has a 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 2.2 (Pointwise hypopthesis 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 ℓ such that $0 \leq \ell(f_{\mathcal{D}}, z) \leq M$). Then with probability $1 - \eta$*

$$R_{gen}[f_{\mathcal{D}}] \leq R_{emp}[f_{\mathcal{D}}] + \sqrt{\eta^{-1} \frac{M^2 + 12Mm\beta_m}{2m}} \tag{3}$$

2.2 Uniform stability

The application of bound (1) to different algorithms $f_1, .., f_T$ with stabilities β_m^t, $t = 1, ..T$, is usually done by using the union bound [36]. Applying theorem 2.1 T times, we get with probability $1 - \eta$,

$$\forall t \in \{1, .., T\}, \quad R_{gen}[f_t] \leq R_{loo}[f_t] + \sqrt{\eta^{-1} T \frac{M^2 + 6Mm\beta_m^t}{2m}} \tag{4}$$

In such situations, we would like to have a dependence in $\ln(T)$ so that we can have large values of T without increasing the bound too much. To this end, we need a stronger notion of stability called uniform stability [4].

Definition 2.3 (Uniform Stability) *An algorithm A has* uniform stability β_m *w.r.t. the loss function ℓ if the following holds*

$$\forall \mathcal{D} \in \mathcal{Z}^m, \ \forall i \in \{1, \ldots, m\}, \ \|\ell(f_{\mathcal{D}}, .) - \ell(f_{\mathcal{D}^{\setminus i}}, .)\|_\infty \leq \beta_m \tag{5}$$

It is easily seen that uniform stability is an upper bound on hypothesis and pointwise hypothesis stability. Uniform stability can be used in the context of regression to get bounds as follows [4]:

Theorem 2.3 *Let $f_{\mathcal{D}}$ be the outcome of an algorithm with uniform stability β_m w.r.t. a loss function ℓ such that $0 \leq \ell(f_{\mathcal{D}}, y) \leq M$, for all $y \in \mathcal{Y}$ and all sets $\mathcal{D}$. Then, for any $m \geq 1$, and any $\eta \in (0, 1)$, the following bound holds with probability $1 - \eta$ over the random draw of the sample $\mathcal{D}$,*

$$R_{gen}[f_{\mathcal{D}}] \leq R_{emp}[f_{\mathcal{D}}] + 2\beta_m + (4m\beta_m + M) \sqrt{\frac{\ln(1/\eta)}{2m}}, \tag{6}$$

and

$$R_{gen}[f_{\mathcal{D}}] \leq R_{loo}[f_{\mathcal{D}}] + \beta_m + (4m\beta_m + M) \sqrt{\frac{\ln(1/\eta)}{2m}}. \tag{7}$$

[1] We adopted the same notation for all notions of stability since it should always be clear from the context which is the referred notion.

The dependence on η is $\sqrt{\ln(1/\eta)}$ 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 – see [4] for more information on how to do this. In the next section, for simplicity we also consider random uniform stability only for regression - classification can again be treated with appropriate changes like in [4].

The notion of uniform stability may appear a little restrictive since inequality (5) has to hold over all training sets $\mathcal{D}$. Note that a weaker notion of stability has been introduced by Kutin and Niyogi [26] with related exponential bounds but its presentation is beyond the scope of this paper.

Example 2.2 (Uniform Stability of regularization methods) *Regularization–based learning algorithms such as Regularization Networks (RN's) [31] and Support Vector Machines (SVM's) [36] are obtained by minimizing the functional*

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

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 $K : X \times X \to \mathbb{R}$ – see, e.g., [38, 16, 22]. A classical 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[2], while SVM's regression uses the loss $\ell(f,y) = |f-y|_\epsilon$, where $|\xi|_\epsilon = |\xi| - \epsilon$ if $|\xi| > \epsilon$, and zero otherwise[3].

It can be shown [4] that for Lipschitz loss functions, the uniform stability 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 2.3 does not give an interesting bound. On the contrary, when λ is large the solution is more stable and Theorem 2.3 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.

To illustrate this point, we plot in figure 2 the absolute difference between the empirical error and the generalization error for SVM regression. The errors are computed on the UCI ionosphere data set that we consider as a regression problem using the ℓ_1 loss function. They are averaged over 500 runs, each run consisting in splitting randomly the data set into one training set of size 35 and one testing set of size 315. This experimental setting is equivalent to considering that $\mathcal{Z}$ is the finite ionosphere data set and $\mathbb{P}$ corresponds to a uniform draw without replacement of m points. The bounds suggest an increase in the difference when λ decreases, which is what we observe. We also show the average generalization error to show that the increase in the difference between generalization and empirical error is *not* due to overfitting.

Figure 2: *Left: Average of the absolute difference between empirical and generalization errors for a gaussian SVM as a function of $1/(2\lambda)$ (log scale) on the ionosphere data set. Right: Average generalization error. The bars represent the 95% confidence interval for each λ of a two sided t-test.*

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 formally to certain neural networks as well as bagging methods. In this section we extend the above notions of stability to randomized learning algorithms and present new results which uncover the link between stability and generalization.

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

[2] In this case, when x is a vector in a Euclidean space with scalar product $(\cdot,\cdot)$ and K is the linear kernel, $K(x,t) = (x,t)$, we recover the ridge regression method – see [22] for the discussion and connection to other methods used in Statistics.

[3] Note that in the statistical learning theory literature [36], 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.

convenience, we will use the shorthand $f_{\mathcal{D},\mathbf{r}}$ as to be 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 in the following examples.

Example 3.1 (Bootstrapping once) *Let $\mathcal{R} = \{1, .., m\}^m$ and define $\mathbb{P}_{\mathbf{r}}$ to be a multinomial distribution with parameters $(\frac{1}{m}, .., \frac{1}{m})$. The random process models the sub-sampling with replacement of m 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 $\mathbf{r} \in \mathcal{R}$ just described. This is bagging using only one bootstrap sampling.*

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

Example 3.2 (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 can be understood as an algorithm A taking a training set and a random vector $\mathbf{r} \in \mathcal{R}$ as inputs, k being here the number of weights of the Network.*

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

- We give stability definitions that correspond to deterministic stability concepts when there is no randomness, i.e. $\mathcal{R}$ 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 $\mathbf{r}$ is decomposed as a vector of random parameters $\mathbf{r} = (\mathbf{r}_1, .., \mathbf{r}_T)$. In this case we write $\mathbf{r} \in \mathcal{R}^T$ drawn from a distribution $\mathbb{P}_{\mathbf{r}^T}$ to indicate the product nature of $\mathbf{r}$, and $\mathbf{r}_t, t = 1, .., T$ are assumed to be random elements drawn from $\mathcal{R}$ *independently* from the same distribution $\mathbb{P}_{\mathbf{r}}$. Notice the slight abuse of notation for simplicity. This is used for example to model the randomization of bagging, where each $\mathbf{r}_t$ corresponds to one random subsampling from the data, and the T subsamples are all drawn independently. We will make use of the following assumption:

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

  This is clearly not restrictive, but instead it is more general. We use this assumption to study the case of bagging below.
- 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 when they introduce random error stability [23]. 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 determinist leave-one-out error [17]. Therefore for each leave-one-out iteration, when we leave one point out we use the same $\mathbf{r}$ - modified only to take into account the removal of a point - for the remaining $m - 1$ points. In the case of bagging we would use the same bootstrap samples that we used when having all m points, without the point left out, 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 3.1, we assume the following:

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

This assumption is not restrictive about the kind of learning methods we can consider: for example both for bagging and for neural networks the same $\mathbf{r}$ (i.e. subsamples for bagging 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 3.1 (Random Hypothesis Stability) *A randomized algorithm* A *has* random hypothesis stability β_m *w.r.t. the loss function* ℓ *if the following holds:*

$$\forall i \in \{1,..,m\}, \mathbf{E}_{\mathcal{D},z,\mathbf{r}}\left[\left|\ell(f_{\mathcal{D},\mathbf{r}},z) - \ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z)\right|\right] \leq \beta_m. \tag{8}$$

Note that the value in the right hand side (r.h.s.) of (8) can vary for different indices i. If $\mathbf{r}$ is fixed then the random hypothesis stability is exactly the same as the hypothesis stability except for one thing: the resulting algorithm need not be symmetric anymore, for example because if we permute the training data different data points are selected when we sample using a fixed $\mathbf{r}$. This means that we cannot apply the results for the case of deterministic algorithms and we have to consider different bounds on the variance of the difference between the generalization and empirical (or leave-one-out) errors. We prove in appendix A the following lemma.

Lemma 3.1 *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:*

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen} - R_{\ell oo})^2\right] \leq \frac{2M^2}{m} + \frac{6M}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[|\ell(f_{\mathcal{D}},z) - \ell(f_{\mathcal{D}^{\setminus i}},z)|\right] \tag{9}$$

Using Tchebychev's inequality, this lemma leads to the following bound:

$$\mathbb{P}_{\mathcal{D}}\left(R_{gen}[f_{\mathcal{D},\mathbf{r}}] - R_{\ell oo}[f_{\mathcal{D},\mathbf{r}}] \geq \epsilon|\mathbf{r}\right) \leq \frac{2M^2}{m\epsilon^2} + \frac{6M\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[\left|\ell(f_{\mathcal{D},\mathbf{r}},z) - \ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z)\right|,\mathbf{r}\right]}{m\epsilon^2} \tag{10}$$

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 fact that $\mathbb{E}_Y\left[\mathbb{E}_X\left[f(X,Y),Y\right]\right] = \mathbb{E}_{X,Y}\left[f(X,Y)\right]$ we derive the following theorem about the generalization and leave-one-out errors of randomized learning methods:

Theorem 3.1 *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* $0 \leq \ell(f,z) \leq M$*, for all* $y \in \mathcal{Y}$*,* $\mathbf{r} \in \mathcal{R}$ *and all sets* $\mathcal{D}$*. Then with probability* $1-\eta$*:*

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \leq R_{\ell oo}[f_{\mathcal{D},\mathbf{r}}] + \sqrt{\eta^{-1}\frac{2M^2 + 6Mm\beta_m}{m}}. \tag{11}$$

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.

Like 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 3.2 (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:*

$$\forall i \in \{1,..,m\}, E_{\mathcal{D}_m,\mathbf{r}}\left|\ell(f_{\mathcal{D},\mathbf{r}},z_i) - \ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z_i)\right| \leq \beta_m. \tag{12}$$

Using the following lemma proved in appendix A,

Lemma 3.2 *For any (non-symmetric) learning algorithm A and loss function ℓ such that $0 \leq \ell(f,z) \leq M$ we have for the empirical error,*

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen} - R_{emp})^2\right] \leq \frac{2M^2}{m} + \frac{12M}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D}}\left[|\ell(f_{\mathcal{D}}, z_i) - \ell(f_{\mathcal{D}^{\setminus i}}, z_i)|\right], \tag{13}$$

We can derive as before the theorem:

Theorem 3.2 *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 $0 \leq \ell(f,z) \leq M$, for all $y \in \mathcal{Y}$, $\mathbf{r} \in \mathcal{R}$ and all sets $\mathcal{D}$. Then with probability $1-\eta$,*

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \leq \mathbb{R}_{emp}[f_{\mathcal{D},\mathbf{r}}] + \sqrt{\eta^{-1}\frac{2M^2 + 12Mm\beta_m}{m}}, \tag{14}$$

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

Definition 3.3 (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 $i = 1, .., m$*

$$\sup_{\mathcal{D},z}\left|\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D},\mathbf{r}}, z)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}}, z)\right]\right| \leq \beta_m \tag{15}$$

Note that this definition is consistent with Definition 2.3 which holds for deterministic symmetric learning algorithms. To link uniform stability to generalization, the following result by McDiarmid [29], see also [11], is central.

Theorem 3.3 (Bounded Difference Inequality) *Let $\mathbf{r}_1, .., \mathbf{r}_T \in \mathcal{R}$ 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*

$$\sup_{\mathbf{r}_1,\ldots,\mathbf{r}_T,\mathbf{r}'_t}|G(\mathbf{r}_1,\ldots,\mathbf{r}_T) - G(\mathbf{r}_1,\ldots,\mathbf{r}_{t-1},\mathbf{r}'_t,\mathbf{r}_{t+1},\ldots,\mathbf{r}_T)| \leq c_t,\ t = 1,\ldots,T. \tag{16}$$

Then, for every $\epsilon > 0$

$$\mathrm{Prob}\left[G(\mathbf{r}_1,\ldots,\mathbf{r}_T) - \mathbf{E}_{\mathbf{r}}\left[G(\mathbf{r}_1,\ldots,\mathbf{r}_T)\right] \geq \epsilon\right] \leq \exp\{-2\epsilon^2/\sum_{t=1}^{T}c_t^2\}. \tag{17}$$

For the next theorem we replace the G of theorem 3.3 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 inequality (17). This is a mild assumption but the bounds below will be interesting only if, for $T \to \infty$, the c_t go to zero at least as $1/\sqrt{T}$. We will see in the next section that for begging $c_t = O(1/T)$.

Theorem 3.4 *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 c_t be a function of t satisfying (16) (G being $\ell(f_{\mathcal{D},\mathbf{r}}, z)$ where $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_T)$) and define $\gamma_T = \max_t c_t$. The following bound holds with probability at least $1-\delta$ with respect to a random sampling of $(\mathcal{D}, \mathbf{r})$:*

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \leq R_{emp}(f_{\mathcal{D},\mathbf{r}}) + 2\beta_m + \left(\frac{M + 4m\beta_m}{\sqrt{2m}} + \sqrt{2T}\gamma_T\right)\sqrt{\log 2/\delta} \tag{18}$$

and, assuming that β_{m-1} the random uniform stability for training sets of size $m-1$ is greater than β_m[4], we have

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

[4]This assumption is natural: when points are added to the training set, the outcome of a learning algorithm is usually more stable. Note that 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.

Proof:

We first prove (18) and then show how to derive (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. To 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

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

Thus, using the definition of γ_T, equation (20) becomes

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

and applying Theorem 3.3 we obtain (note that $\mathcal{D}$ is independent of $\mathbf{r}$)

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

We also have

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

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

$$K(\mathcal{D},\mathbf{r}) - \mathbf{E}_{\mathbf{r}}K(\mathcal{D},\mathbf{r}) \leq \sqrt{2T}\gamma_T\sqrt{\log(1/\eta)}. \tag{20}$$

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

$$\begin{aligned} \mathbf{E}_{\mathcal{D},\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] &= \mathbf{E}_{\mathcal{D}}\left[\frac{1}{m}\sum_{i=1}^m G(\mathcal{D},z_i) - \mathbf{E}_z\left[G(\mathcal{D},z)\right]\right] && (21)\\ &= \frac{1}{m}\sum_{i=1}^m \mathbf{E}_{\mathcal{D}}\left[G(\mathcal{D},z_i)\right] - \mathbf{E}_{\mathcal{D},z}\left[G(\mathcal{D},z)\right] && (22)\\ &\overset{(a)}{\leq} 2\beta_m + \frac{1}{m}\sum_{i=1}^m \mathbf{E}_{\mathcal{D}}\left[G(\mathcal{D}^{\setminus i},z_i)\right] - \mathbf{E}_{\mathcal{D}^{\setminus i},z}\left[G(\mathcal{D}^{\setminus i},z)\right] && (23)\\ &\overset{(b)}{\leq} 2\beta_m && (24) \end{aligned}$$

where (a) is derived from the fact that the algorithm has a 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 3.3 to $\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right]$. To this aim, we bound (recall that $\mathcal{D}^i = \mathcal{D}^{\setminus i} \cup z'$):

$$\begin{aligned} &\left|\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] - \mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D}^i,\mathbf{r})\right]\right| = \\ &\left|\underbrace{\frac{1}{m}\left(\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D},\mathbf{r}},z_i)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^i,\mathbf{r}},z')\right]\right)}_{(a)} + \frac{1}{m}\sum_{j=1,j\neq i}^m \underbrace{\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D},\mathbf{r}},z_j)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z_j)\right]}_{(b)}\right. \\ &\left. + \frac{1}{m}\sum_{j=1,j\neq i}^m \underbrace{\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z_j)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^i,\mathbf{r}},z_j)\right]}_{(c)} + \underbrace{\mathbf{E}_{\mathbf{r}}\left[\mathbf{E}_z\left[\ell(f_{\mathcal{D},\mathbf{r}},z) - \ell(f_{\mathcal{D}^i,\mathbf{r}},z)\right]\right]}_{(d)}\right| \end{aligned} \tag{25}$$

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:

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

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

$$\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] \leq 2\beta_m + \frac{M+4m\beta_m}{\sqrt{2m}}\sqrt{\log(1/\eta)} \tag{26}$$

Observe that inequalities (20) and (26) hold simultaneously with probability at least $1-2\eta$. The result follows by combining those inequalities and setting $\eta = \delta/2$.

The proof of equation (19) follows the same reasoning except that the chain of equations (21-24) and (25) are different. We have:

$$\begin{aligned}
\mathbf{E}_{\mathcal{D},\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] &= \mathbf{E}_{\mathcal{D}}\left[\frac{1}{m}\sum_{i=1}^{m} G(\mathcal{D}^{\setminus i}, z_i) - \mathbf{E}_z\left[G(\mathcal{D},z)\right]\right] &(27)\\
&= \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[G(\mathcal{D}^{\setminus i},z)\right] - \mathbf{E}_{\mathcal{D},z}\left[G(\mathcal{D},z)\right] &(28)\\
&\leq \beta_m &(29)
\end{aligned}$$

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 $j \neq i$),

$$\begin{aligned}
&\left|\mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D},\mathbf{r})\right] - \mathbf{E}_{\mathbf{r}}\left[K(\mathcal{D}^i,\mathbf{r})\right]\right| = \\
&\Bigg|\underbrace{\frac{1}{m}\left(\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z_i)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i},\mathbf{r}},z')\right]\right)}_{(a)} + \frac{1}{m}\sum_{j=1,j\neq i}^{m}\underbrace{\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus j},\mathbf{r}},z_j)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i,j},\mathbf{r}},z_j)\right]}_{(b)} \\
&+\frac{1}{m}\sum_{j=1,j\neq i}^{m}\underbrace{\mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{\setminus i,j},\mathbf{r}},z_j)\right] - \mathbf{E}_{\mathbf{r}}\left[\ell(f_{\mathcal{D}^{i\setminus j},\mathbf{r}},z_j)\right]}_{(c)} + \underbrace{\mathbf{E}_{\mathbf{r}}\left[\mathbf{E}_z\left[\ell(f_{\mathcal{D},\mathbf{r}},z) - \ell(f_{\mathcal{D}^i,\mathbf{r}},z)\right]\right]}_{(d)}\Bigg| \qquad (30)
\end{aligned}$$

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

Like in the deterministic case discussed in section 2, similar results can be given for classifi cation like in [4].

4 Stability of bagging and subagging

Bagging [5] and subagging [21] are randomized algorithms which work by averaging the solutions of a learning algorithm trained several 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 "modifi es" the stability of the base (underline) learning algorithm. First let us present more formally what we mean by bagging.

4.1 Bagging

Bagging consists in 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 $t = 1,..,T$, where the $\mathbf{r}_t \in \mathcal{R} = \{1,..,m\}^m$ 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 3.1). Such random variables have a multinomial distribution with parameters $(\frac{1}{m},..,\frac{1}{m})$. The overall bagging model can thus be written as:

$$F_{\mathcal{D},\mathbf{r}} = \frac{1}{T}\sum_{t=1}^{T} f_{\mathcal{D}(\mathbf{r}_t)}. \tag{31}$$

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[5]. 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 already heavy notation, we suppress the symbol $\mathbf{s}_t$.

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 $A-$lipschitzian w.r.t. its first variable. Let $F_{\mathcal{D}}$ 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 $F_{\mathcal{D}}$ with respect to ℓ is bounded by:*

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

where $d(\mathbf{r})$ is the number of independent coordinates of the vector $\mathbf{r}$, that is the number of distinct sampled points.

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

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

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

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

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

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

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

$$\mathbf{E}_{\mathcal{D},z} \left[I(\mathcal{D}, z) \right] \leq A \mathbf{E}_{\mathbf{r},\mathcal{D},x} \left[|\Delta(\mathcal{D}(\mathbf{r}), x)| \mathbf{1}_{i \in \mathbf{r}} \right] = A \mathbf{E}_{\mathbf{r}} \left[\mathbf{E}_{\mathcal{D},x} \left[|\Delta(\mathcal{D}(\mathbf{r}), x)| \right] \mathbf{1}_{i \in \mathbf{r}} \right] = A \mathbf{E}_{\mathbf{r}} \left[\gamma_{d(\mathbf{r})} \mathbf{1}_{i \in \mathbf{r}} \right] \quad (32)$$

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 conclude the proof for hypothesis stability. The proof for pointwise stability is exactly the same except that in equation (32) there is no expectation w.r.t. z and z is replaced by z_i. $\Diamond$

[5]This means that if for example the underline 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.

The bounds we just proved depend on the quantities $\mathbb{P}_{\mathbf{r}}[d(\mathbf{r}) = k]$, where, we recall that $d(\mathbf{r})$ is the number of distinct sampled points. It can be shown, for example by applying theorem 3.3, that the random variable $d(\mathbf{r})$ is sharply concentrated around its mode which is for $k = 0.632m$. For that reason, in what follows we will assume that the previous bounds can be approximately rewritten as:

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

Then, if $A = 1$ the bounds derived for the bagging predictor are better than those of the single predictor trained on the whole training set which use the hypothesis stability for the single predictor trained on the whole training set. Notice also that .632 is the probability that the bootstrapped set will contain a specifi c (any) point, also used to justify the .632 bootstrap error estimates [15].

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

Proposition 4.2 (Random hypothesis stability of bagging for classification) *Let $F_{\mathcal{D}}$ 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 $F_{\mathcal{D}}$ 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, $A = 1$. 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

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

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

Example 4.1 (k-NN) *As previously seen, k-NN has a hypothesis stability equal to $\frac{k}{m}$ such that bagging k-NN has a stability with respect to classification loss bounded by:*

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

which is approximately $\frac{2k}{m}$. So bagging does not improve stability, which is also experimentally verified by Breiman [5].

The next proposition establishes the link between the uniform stability of bagging and that of the base learning algorithm for regression. As before, classifi cation can be treated similarly like in [4].

Proposition 4.3 (Random uniform stability of bagging for regression) *Assume that the loss ℓ is A-lipschitzian with respect to its first variable. Let $F_{\mathcal{D}}$ 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 $F_{\mathcal{D}}$ with respect to ℓ is bounded by:*

$$\beta_m \leq A \sum_{k=1}^{m} \frac{k \gamma_k}{m} \mathbb{P}_{\mathbf{r}} \left[d(\mathbf{r}) = k \right]. \tag{33}$$

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:

$$\beta_m \leq A \sup_{\mathcal{D},x} \{\mathbf{E}_{\mathbf{r}} [\Delta(\mathcal{D}(\mathbf{r}), x) \mathbf{1}_{i \in \mathbf{r}}]\}$$

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

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

Now we observe that

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

Placing this bound in the previous one gives

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

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

Example 4.2 (SVM regression) *We have seen in example 2.2 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.*

To illustrate the last example, we used the same experimental setting In example 2.2 and compared the average of the absolute difference between the empirical and the generalization error of a single SVM regression and that of a bagged SVM using the same λ for all bootstrap sets. Figure 3 reports the results.

Figure 3: *Average of the absolute difference between empirical and generalization errors for gaussian SVMs with $1/(2\lambda) = 0.01, .., 100$ (log scale) on the ionosphere data set. The bars represent the 95% confidence interval for each λ of a two sided t-test. The dotted line corresponds to a single SVM trained on the whole training set. The plain line is the result of bagging 20 SVMs, each SVM being trained with the same λ.*

4.2 Subagging

Subagging is a variation of bagging where the sets $\mathcal{D}(\mathbf{r}_t)$, $t = 1, \ldots, T$ 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 taken directly which gives the following upper bounds on stability for subagging:

Proposition 4.4 ((Hypothesis, Pointwise hypothesis, Uniform) Stability of subagging for regression) *Assume that the loss ℓ is A-lipschitzian w.r.t. its first variable. Let $F_{\mathcal{D}}$ be the outcome of a subagging 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 subagging is done by sampling p points without replacement. Then the random uniform (resp. hypothesis or pointwise hypothesis) stability β_m of $F_{\mathcal{D}}$ w.r.t. ℓ is bounded by:*

$$\beta_m \leq A \gamma_p \frac{p}{m}$$

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

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

We can now prove bounds on the performance of bagging and subagging. We present the following theorems for subagging but the same statements holds true for bagging where $\frac{p\gamma_p}{m}$ 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 4.1 *Assume that the loss ℓ is A-lipschitzian w.r.t. its first variable. Let $F_{\mathcal{D}}$ be the outcome of a subagging algorithm. Assume subagging 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 $1 - \delta$*

$$R_{gen}(F_{\mathcal{D}}) \leq R_{\ell oo}(F_{\mathcal{D}}) + \sqrt{\delta^{-1} \frac{2M^2 + 6MAp\gamma_p}{m}} \tag{34}$$

$$R_{gen}(F_{\mathcal{D}}) \leq R_{emp}(F_{\mathcal{D}}) + \sqrt{\delta^{-1} \frac{2M^2 + 6MAp\gamma_p}{m}} \tag{35}$$

Proof:

The inequalities follows from by plugging the result in Proposition 4.4 in Theorems 3.1 and 3.2 respectively. $\Diamond$

Note that, as in proposition 4.2, the same result holds for classification if we set $A = 2$ and $M = 1$.
The following theorem holds for regression. The extension to the case of classification can be done again as in [4].

Theorem 4.2 *Under the same conditions above with hypothesis stability quantities replaced by the uniform stability, the following bounds hold separately with probability at least $1 - \delta$ in the case of regression*

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \leq R_{\ell oo}(f_{\mathcal{D},\mathbf{r}}) + \frac{Ap\gamma_p}{m} + \left(\frac{M + 4A(m/m - 1)p\gamma_p}{\sqrt{2m}} + \frac{\sqrt{2}AM}{\sqrt{T}} \right) \sqrt{\log(2/\delta)} \tag{36}$$

and

$$R_{gen}(f_{\mathcal{D},\mathbf{r}}) \leq R_{emp}(f_{\mathcal{D},\mathbf{r}}) + 2\frac{Ap\gamma_p}{m} + \left(\frac{M + 4Ap\gamma_p}{\sqrt{2m}} + \frac{\sqrt{2}AM}{\sqrt{T}} \right) \sqrt{\log 2/\delta} \tag{37}$$

Proof:

We recall that $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_T)$ and introduce the notation $\mathbf{r}^t = (\mathbf{r}_1, \ldots, \mathbf{r}_{t-1}, \mathbf{r}', \mathbf{r}_{t+1}, \ldots, \mathbf{r}_T)$. Note that

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

Thus, the constant γ_T which appears in Theorem 3.4 is bounded as

$$\gamma_T = \sup_{\mathbf{r},\mathbf{r}'_t} |\ell(f_{\mathcal{D},\mathbf{r}}, z) - \ell(f_{\mathcal{D},\mathbf{r}^t}, z)| \leq \frac{A}{T} M.$$

The result then follows by using this theorem and Proposition 4.4. $\Diamond$

The results of this section show the following characteristics of bagging and subagging:

- The main effects of bagging and subagging are on the stability, as shown in sections 4.1 and 4.2. If, as $m \to \infty$, $\frac{p\gamma_p}{\sqrt{m}} \to 0$, the empirical error of subagging converges to the expected error. Similar asymptotic results have been shown for the bootstrap error in [32]. In the extreme case, fixing p as $m \to \infty$ implies that the empirical error converges to the expected one. The convergence is in probability when hypothesis stability is used and almost surely for uniform stability.
- Theorem 4.2 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 [5].

4.4 Bias/Variance decomposition and stability

Finally we argue in this section that the bias/variance considerations used by Breiman [5] to motivate his original work on bagging can be understood more precisely by invoking the stability concepts we developed here. We first state two results which are important for this discussion.

Theorem 4.3 (Devroye [10]) *Let $f(x_1, \ldots, x_m)$ be a random variable depending on m i.i.d. random variables $x_1, .., x_m$ with the property that if $\hat{x}_i$ is a replicate of x_i, $\forall i \in \{1, \ldots, m\}$:*

$$\sup_{x_1,\ldots,x_m,\hat{x}_i} |f(x_1, \ldots, x_m) - f(x_1, \ldots, x_{i-1}, \hat{x}_i, x_{i+1}, \ldots, x_m)| \leq c_i$$

Then we have the following bound on the variance of f:

$$Var\left(f(x_1, \ldots, x_m)\right) \leq \frac{1}{4}\sum_{i=1}^{m} c_i^2$$

Theorem 4.4 (Steele [34]) *Under the same hypotheses of the theorem above:*

$$Var\left(f(x_1, .., x_m)\right) \leq \frac{1}{2}\sum_{i=1}^{m} \mathbb{E}_{x_1,..,x_m,\hat{x}_i}\left[\left(f(x_1, \ldots, x_m) - f(x_1, \ldots, x_{i-1}, \hat{x}_i, x_{i+1}, \ldots x_m)\right)^2\right]$$

The bias/variance discussion introduced by Breiman concerned an ideal bagging predictor defined as the average over all possible training sets $\mathcal{D}$: $F(x) = \mathbf{E}_{\mathcal{D}}\left[f_{\mathcal{D}}(x)\right]$, which he assumed to be close to the bagging predictor. He showed that

$$\mathbf{E}_{\mathcal{D}}\left[R(f_{\mathcal{D}})\right] = R(\mathbf{E}_{\mathcal{D}}\left[f_{\mathcal{D}}\right]) + \mathbf{E}_{\mathcal{D},x}\left[\left(f_{\mathcal{D}} - \mathbf{E}_{\mathcal{D}}\left[f_{\mathcal{D}}\right]\right)^2\right].$$

We then see that, if for example the base classifier has uniform stability β_m, Theorem 4.3 gives

$$\mathbf{E}_{\mathcal{D}}\left[R(f_{\mathcal{D}})\right] \leq R(\mathbf{E}_{\mathcal{D}}\left[f_{\mathcal{D}}\right]) + m\beta_m^2.$$

Thus, if $f_{\mathcal{D}}$ is unstable, its average error will be much larger that the error of $\mathbf{E}_{\mathcal{D}}\left[f_{\mathcal{D}}\right]$ and the use of bagging should greatly improve performance. However, note that this reasoning is not rigorous because the bagging predictor is just an estimates of $\mathbf{E}_{\mathcal{D}}\left[f_{\mathcal{D}}\right]$. We now develop the discussion more formally to include this predictor in Eq. (31). Again when ℓ is the square loss, it is easy to see that

$$\mathbf{E}_{\mathcal{D},\mathbf{r}}\left[R(F_{\mathcal{D},\mathbf{r}})\right] = R(\bar{F}) + \mathbf{E}_x\left[\text{Variance1}(x)\right] + \mathbf{E}_{\mathcal{D},x}\left[\text{Variance2}(\mathcal{D}, x)\right] \tag{38}$$

where we used the notation $\bar{F} = \mathbf{E}_{\mathcal{D},\mathbf{r}}\left[F_{\mathcal{D},\mathbf{r}}\right] = \mathbf{E}_{\mathcal{D},\mathbf{r}}\left[f_{\mathcal{D}(\mathbf{r})}\right]$ and:

$$\text{Variance1}(x) = \mathbf{E}_{\mathcal{D}}\left[\left(\mathbf{E}_{\mathcal{D},\mathbf{r}}\left[F_{\mathcal{D},\mathbf{r}}(x)\right] - \mathbf{E}_{\mathbf{r}}\left[F_{\mathcal{D},\mathbf{r}}(x)\right]\right)^2\right] \tag{39}$$

$$\text{Variance2}(\mathcal{D}, x) = \mathbf{E}_{\mathbf{r}}\left[\left(\mathbf{E}_{\mathbf{r}}\left[F_{\mathcal{D},\mathbf{r}}(x)\right] - F_{\mathcal{D},\mathbf{r}}(x)\right)^2\right] \tag{40}$$

We note that $\text{Variance1}(x)$ can be bounded by using Theorem 4.3 as

$$\text{Variance1}(x) \leq m(\beta_m^u)^2$$

where β_m^u is the uniform stability of $F_{\mathcal{D},\mathbf{r}}$. On the other hand, Theorem 4.4 implies

$$\text{Variance1}(x) \leq 2mM\beta_m^h$$

where β_m^h is the hypothesis stability. Next we note that by the same theorem

$$\text{Variance2}(\mathcal{D}, x) \leq \frac{2M^2}{T}.$$

Putting all together we obtain

$$\mathbf{E}_{\mathcal{D},\mathbf{r}}\left[R(F_{\mathcal{D},\mathbf{r}})\right] \leq R(\bar{F}) + m\min((\beta_m^{(u)})^2, 2M\beta_m^{(h)}) \; + \frac{2M^2}{T}.$$

Thus, the reasoning Breiman has followed in his original work and which was mainly about stability can here be understood more formally by using the well defined notion of stability that has already been used to understand generalization properties. In particular, if $F_{\mathcal{D},\mathbf{r}}$ is the subagging algorithm the variance is controlled by p, the number of samples used by the underline learning algorithm. By propositions 4.4, we have $\beta_m \leq \gamma_p p/m$, with γ_p being the stability of the underline algorithm. So, if γ_m is sublinear in m, the variance always decreases with p.

So far we dealt only with the variance analysis. The bias is of $\bar{F}$ is defined by $\mathrm{Bias}(\bar{F}) = \mathbf{E}_x\left[(f_0(x) - \bar{F}(x))^2\right]$, where f_0 is the regression function, $f_0 = \mathbf{E}_y[y;x]$, the conditional average of the output. We have:

$$R(\bar{F}) = R(f_0) + \mathrm{Bias}(\bar{F})$$

We conclude that, for $T \gg M^2$

$$\mathbf{E}_{\mathcal{D},\mathbf{r}}\left[R(F_{\mathcal{D},\mathbf{r}})\right] \approx R(f_0) + \mathrm{Bias}(p) + \mathrm{Variance}(p)$$

We expect the variance to increase with p and the bias to decrease. Thus, according to this analysis the optimal value of p is the minimizer of the r.h.s. of the above equation. Of course, to proceed further in this study we need an estimate for the bias.

5 Conclusion

We presented a theory of random stability for randomized learning methods. This is an extension of the existing theory about the stability and generalization performance of deterministic (symmetric) learning methods [4]. We then applied the theory to the case of bagging. The bounds say that both the stability of the learning methods used as well as the size of the bootstrap subsamples (i.e. the percentage of the original training set used) are important for estimating the difference between empirical and expected error for bagging. An important practical consequence of the bounds is that for small subsampling size, the empirical error of subagging is a good indicator of the expected error for this type of bagging as shown experimentally in [17]. Finally, in this paper we mentioned two open questions: a) how to extend the results in section 4 to the case where the base learning methods handle multiple copies of a training point not as a single point but as many; b) how to use the results presented here to study *non*-asymptotically the quality of bootstrap error estimates [13, 15]. Notice that to answer (b) we first need to answer (a), since the bootstrap error estimate assumes multiple points are treated as such.

References

[1] 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.

[2] E. Bauer and R. Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants. *Machine Learning*, 36:105–142, 1999.

[3] S. Boucheron, G. Lugosi, and P. Massart. A sharp concentration inequality with applications. *Random Structures and Algorithms*, 16:277–292, 2000.

[4] O. Bousquet and A. Elisseeff. Stability and generalization. To be published in *Journal of Machine Learning Research*, 2001.

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

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

[7] L. Breiman. Arcing Classifiers. *Annals of Statistics*, Volume 26,1998.

[8] L. Breiman. Some Infinite Theory for Predictor Ensemble. Univerisity of California, Berkeley, *Tech. Report* 579, 2000.

[9] P. Bhulmann and B. Yu. Explaining bagging. Available at: www.stat.Berkeley.EDU/users/binyu/publications.html, 2000.

[10] L. Devroye. *Exponential inequalities for sums of bounded random variables.* In G. Roussas, editor, Nonparametric Functional Estimation and Related Topics, p. 31–44. NATO ASI Series, Kluwer Academic Publishers, Dordrecht, 1991.

[11] L. Devroye, L. Gy¨orfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Number 31 in Applications of mathematics. Springer, New York, 1996.

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

[13] B. Efron Estimating the error rate of a prediction rule: improvement on cross-validation *Journal of the American Statistical Association*, Vol. 78, No. 382, p. 316-331, June 1983.

[14] B. Efron and R. Tibshirani *An Introduction to the Bootstrap* New York, Chapman and Hall, 1993.

[15] B. Efron and R. Tibshirani Improvements on cross-validation: the .632+ bootstrap method *Journal of the American Statistical Association*, Vol. 97, No. 438, p. 548–560, June 1997.

[16] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines.

[17] T. Evgeniou, M. Pontil, and A. Elisseeff. Leave one out error, stability, and generalization of voting combinations of classifiers *Machine Learning*, 2003 (to appear).

[18] Y. Freund and R. Schapire. Discussion of the paper "Arcing Classifier" by Leo Breiman. *Annals of Statistics*,26(3):824–832, 1998.

[19] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to Boosting. *Journal of Computer and System Sciences*,55(1):119–139, 1997.

[20] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Tech. report, Department of Statistics, Stanford University, 1998.

[21] J. Friedman and P. Hall. On Bagging and Non-linear Estimation. Preprint,2000.

[22] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data Mining, Inference, and Prediciton*. Springer Series in Statistics, 2002.

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

[24] R. Kohavi A study of cross-validation and bootstrap for accuracy assessment anf model selection International Joint Conference in Artificial Intelligence, 1995.

[25] R. Kohavi and D. Wolpert. Bias plus Variance Decomposition for Zero-one Loss Functions. In *Machine Learning: Proceedings of the Thirteenth International Conference*, p. 275–283, 1996.

[26] S. Kutin and P. Niyogi The interaction of stability and weakness in AdaBoost Technical Report TR-2001-30, Computer Science Department, University of Chicago, 2001.

[27] M. LeBlanc and R. Tibshirani. Combining estimates in regression and classification. *Journal of the American Statistical Association*, Vol. 91, No. 436, p. 1641–1650, 1996.

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

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

[30] J. Quinlan. Baggin, Boosting and C4.5. In *Proceedings of the Thirteenth National Conference on*

[31] T. Poggio and F. Girosi. Networks for Approximation and Learning. Proceedings of the IEEE*, 78 (9), 1990.*

[32] J. Shao. Bootstrap Model Selection. Journal of the American Statistical Associacion*, Vol. 91, No. 434, p. 655– 665, June 1996.*

[33] R. Shapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods.

[34] J.M. Steele. An Efron-Stein inequality for nonsymmetric statistics. Annals of Statistics, 14:753-758, 1986. The Annals of Statistics*, 1998.*

[35] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. Theory Probab. Appl.*, 16:264– 180, 1971.*

[36] V. N. Vapnik. Statistical Learning Theory. *Wiley, New York, 1998.*

[37] D. Wolpert. Stacked Generalization. Neural Networks*, 5, 241– 359, 1992.*

[38] G. Wahba. Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.*

A Proof of Lemma 3.1 and 3.2

Lemma A.1 *For any (non-symmetric) learning algorithm A and loss function ℓ such that $0 \leq \ell(f,z) \leq M$ we have for the empirical error,*

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen} - R_{emp})^2\right] \leq \frac{2M^2}{m} + \frac{12M}{m^2}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D}}\left[|\ell(f_{\mathcal{D}}, z_i) - \ell(f_{\mathcal{D}^{\setminus i}}, z_i)|\right], \tag{41}$$

and for the leave-one-out error,

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen} - R_{loo})^2\right] \leq \frac{2M^2}{m} + \frac{6M}{m^2}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[|\ell(f_{\mathcal{D}}, z) - \ell(f_{\mathcal{D}^{\setminus i}}, z)|\right] \tag{42}$$

The proof of the lemma follows directly the proof that has been given in [4]. 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 notations: we will denote by

$$\ell_{ij}(z,z',z'') = \ell(f_{\mathcal{D}_{ij}(z,z')},z'') \tag{43}$$

the loss of an algorithm A trained on

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

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 $\{1,..,m\}$, we use the notation $\ell(z)$ to denote $\ell_{ij}(z_i,z_j,z)$ for all i and j in $\{1,..,m\}$. 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. 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 $\mathbf{E}_{\mathcal{D},z}\left[|\ell_{ij}(z,z_j,z_i)-\ell(z_i)|\right]$, 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 A.2 *Let A be a (non-symmetric) algorithm and let ℓ be as defined in (43), we have: $\forall (i,j) \in \{1,..,m\}^2$*

$$\mathbf{E}_{\mathcal{D},z}\left[|\ell(z)-\ell_{ij}(z_j,z_i,z)|\right] \leq \frac{3}{2}\left(\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell(z)\right|\right] + \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z_i,z',z)-\ell(z)\right|\right]\right) \tag{44}$$

Proof:

We have:

$$\begin{aligned}\mathbf{E}_{\mathcal{D},z}\left[|\ell(z)-\ell_{ij}(z_j,z_i,z)|\right] &\leq \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell(z)-\ell_{ij}(z',z_j,z)\right|\right] \\ &+ \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell_{ij}(z',z_i,z)\right|\right] + \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_i,z)-\ell_{ij}(z_j,z_i,z)\right|\right]\end{aligned} \tag{45}$$

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.

$$\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell_{ij}(z',z_i,z)\right|\right] = \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell(z)-\ell_{ij}(z_i,z',z)\right|\right] \tag{46}$$

$$\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_i,z)-\ell_{ij}(z_j,z_i,z)\right|\right] = \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell(z)\right|\right] \tag{47}$$

which gives the following result:

$$\mathbf{E}_{\mathcal{D},z}\left[|\ell(z)-\ell_{ij}(z_j,z_i,z)|\right] \leq 2\mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z',z_j,z)-\ell(z)\right|\right] + \mathbf{E}_{\mathcal{D},z}\left[\left|\ell_{ij}(z_i,z',z)-\ell(z)\right|\right] \tag{48}$$

If instead of (45) we used the following decomposition,

$$\begin{aligned}\mathbf{E}_{\mathcal{D},z}\left[|\ell(z)-\ell_{ij}(z_j,z_i,z)|\right] &\leq \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell(z)-\ell_{ij}(z_i,z',z)\right|\right] \\ &+ \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z_i,z',z)-\ell(g(z_j,z'),z)\right|\right] + \mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell(g(z_j,z'),z)-\ell_{ij}(z_j,z_i,z)\right|\right]\end{aligned} \tag{49}$$

It would have led to:

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

Averaging the this inequality with (48), we get the final result. ◊

Note that the quantity appearing in the r.h.s. of (44) can be bounded by different quantities related to pointwise hypothesis stability or to hypothesis stability. We have indeed:

$$\mathbf{E}_{\mathcal{D},z}\left[|\ell(z)-\ell_{ij}(z_j,z_i,z)|\right] \leq 3\left(\mathbf{E}_{\mathcal{D},z}\left[|\ell_{ij}(z,z_j,z_i)-\ell(z_i)|\right] + \mathbf{E}_{\mathcal{D},z}\left[|\ell_{ij}(z_i,z,z_j)-\ell(z_j)|\right]\right) \tag{50}$$

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

$$\mathbf{E}_{\mathcal{D},z}\left[|\ell(z)-\ell_{ij}(z_j,z_i,z)|\right] \leq 3\left(\mathbf{E}_{\mathcal{D},z}\left[|\ell_{ij}(\emptyset,z_j,z)-\ell(z)|\right] + \mathbf{E}_{\mathcal{D},z}\left[|\ell_{ij}(z_i,\emptyset,z)-\ell(z)|\right]\right) \tag{51}$$

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 [4]:

Lemma A.3 *For any (non-symmetric) learning algorithm A, we have:*

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen}-R_{emp})^2\right] \leq \frac{1}{m^2}\sum_{i\neq j=1}^{m}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right] - \frac{2}{m^2}\sum_{i\neq j=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right] + \frac{1}{m^2}\sum_{i\neq j=1}^{m}\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)\ell(z_j)\right] + \frac{1}{m^2}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right] - 2\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right] + \mathbf{E}_{\mathcal{D}}\left[\ell(z_i)^2\right] \quad (52)$$

and

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen}-R_{\ell oo})^2\right] \leq \frac{1}{m^2}\sum_{i\neq j=1}^{m}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right] - \frac{2}{m^2}\sum_{i\neq j=1}^{m}\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell_{ij}(\emptyset,z_j,z_i)\right] + \frac{1}{m^2}\sum_{i\neq j=1}^{m}\mathbf{E}_{\mathcal{D}}\left[\ell_{ij}(\emptyset,z_j,z_i)\ell_{ij}(z_i,\emptyset,z_j)\right] + \frac{1}{m^2}\sum_{i=1}^{m}\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right] - 2\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell_{ij}(\emptyset,z_j,z_i)\right] + \mathbf{E}_{\mathcal{D}}\left[\ell_{ij}(\emptyset,z_j,z_i)^2\right] \quad (53)$$

Proof:

We have

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

and also

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

and also

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

Also we have

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

and

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

which concludes the proof. ◊

Let's first formulate the first inequality of Lemma (A.3) as

$$\mathbf{E}_{\mathcal{D}}\left[(R_{gen}-R_{emp})^2\right] \leq \frac{1}{m^2}\sum_{i\neq j=1}^{m}\underbrace{\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]}_{I} + \frac{1}{m^2}\sum_{i\neq j=1}^{m}\underbrace{\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)\ell(z_j)\right]-\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]}_{J} + \frac{1}{m^2}\sum_{i=1}^{m}\underbrace{\mathbf{E}_{\mathcal{D},z,z'}\left[\ell(z)\ell(z')\right]-2\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]+\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)^2\right]}_{K} \quad (54)$$

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:

$$\begin{aligned} I = &\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell(z)-\ell_{ij}(z,z_j,z))\ell(z')\right] \\ &+\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z_j,z)-\ell_{ij}(z',z_j,z))\ell(z')\right] \\ &+\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell(z')-\ell_{ij}(z',z_j,z'))\ell_{ij}(z',z_j,z)\right] \end{aligned} \quad (55)$$

Thus,

$$|I| \leq 3M\mathbf{E}_{\mathcal{D},z,z'}\left[\left|\ell_{ij}(z',z_j,z)-\ell(z)\right|\right] \quad (56)$$

Next we rewrite J as

$$\mathbf{E}_{\mathcal{D}}\left[\ell(z_i)\ell(z_j)\right]-\mathbf{E}_{\mathcal{D},z}\left[\ell(z)\ell(z_i)\right]=\mathbf{E}_{\mathcal{D},z,z'}\left[\ell_{ij}(z,z',z)\ell_{ij}(z,z',z')-\ell(z)\ell(z_i)\right] \quad (57)$$

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

$$J=\mathbf{E}_{\mathcal{D},z,z'}\left[\ell_{ij}(z,z',z)\ell_{ij}(z,z',z')-\ell_{ij}(z',z_i,z)\ell_{ij}(z',z_i,z')\right] \quad (58)$$

where we renamed z_i as z' and z_j as z_i in the second term. Using equation 50, we have:

$$\begin{aligned} J = &\underbrace{\mathbf{E}_{\mathcal{D},z,z'}\left[\ell_{ij}(z,z',z)\ell_{ij}(z,z',z')-\ell_{ij}(z_i,z',z)\ell_{ij}(z',z_i,z')\right]}_{J_1} \\ &+3M\left(\mathbf{E}_{\mathcal{D},z}\left[|\ell_{ij}(z,z_j,z_i)-\ell(z_i)|\right]+\mathbf{E}_{\mathcal{D},z}\left[|\ell_{ij}(z_i,z,z_j)-\ell(z_j)|\right]\right) \end{aligned} \quad (59)$$

Let us focus on J_1, we have:

$$\begin{aligned} J_1 = &\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z',z')-\ell_{ij}(z,z_i,z')\ell_{ij}(z,z',z)\right] \\ &+\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z',z)-\ell_{ij}(z_i,z',z))\ell_{ij}(z,z_i,z')\right] \\ &+\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z_i,z')-\ell_{ij}(z',z_i,z'))\ell_{ij}(z_i,z',z)\right] \end{aligned} \quad (60)$$

and

$$\begin{aligned} J_1 = &\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z_i,z_j,z_j)-\ell_{ij}(z_i,z,z_j))\ell_{ij}(z_i,z_j,z_i)\right] \\ &+\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z_i,z_j,z_i)-\ell_{ij}(z,z_j,z_i))\ell_{ij}(z,z_i,z_j)\right] \\ &+\mathbf{E}_{\mathcal{D},z,z'}\left[(\ell_{ij}(z,z_j,z_i)-\ell_{ij}(z_i,z_j,z_i))\ell_{ij}(z_j,z_i,z)\right] \end{aligned} \quad (61)$$

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,

$$|J_1| \leq 2M\mathbf{E}_{\mathcal{D},z}\left[|\ell_{ij}(z,z_j,z_i)-\ell(z_i)|\right] + M\mathbf{E}_{\mathcal{D},z,z'}\left[|\ell_{ij}(z_i,z,z_j)-\ell(z_j)|\right] \quad (62)$$

Summing inequalities (56) and with the inequality on J derived from (62) and (59), we obtain

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

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:

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

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

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

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 $\ell_{ij}(z, z', z)$ 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.