Journal of Machine Learning Research 25 (2024) 1-35 Submitted 4/23; Revised 12/23; Published 4/24

Bagging Provides Assumption-free Stability

Jake A. Soloff SOLOFF@UCHICAGO.EDU
Rina Foygel Barber RINA@UCHICAGO.EDU
Department of Statistics
University of Chicago
5747 S Ellis Ave
Chicago, IL 60637, USA

Rebecca Willett WILLETT@UCHICAGO.EDU
Departments of Statistics and Computer Science
University of Chicago
5735 S Ellis Ave
Chicago, IL 60637, USA

Editor: Samory Kpotufe

Abstract

Bagging is an important technique for stabilizing machine learning models. In this paper, we derive a finite-sample guarantee on the stability of bagging for any model. Our result places no assumptions on the distribution of the data, on the properties of the base algorithm, or on the dimensionality of the covariates. Our guarantee applies to many variants of bagging and is optimal up to a constant. Empirical results validate our findings, showing that bagging successfully stabilizes even highly unstable base algorithms.

Keywords: stability, bagging, bootstrap methods, distribution-free

1. Introduction

Algorithmic stability—that is, how perturbing training data influences a learned model—is fundamental to modern data analysis. In learning theory, certain forms of stability are necessary and sufficient for generalization (Bousquet and Elisseeff, 2002; Poggio et al., 2004; Shalev-Shwartz et al., 2010). In model selection, stability measures can reliably identify important features (Meinshausen and Bühlmann, 2010; Shah and Samworth, 2013; Ren et al., 2023). In scientific applications, stable methods promote reproducibility, a prerequisite for meaningful inference (Yu, 2013). In distribution-free prediction, stability is a key assumption for the validity of jackknife (that is, leave-one-out cross-validation) prediction intervals (Barber et al., 2021; Steinberger and Leeb, 2023).

Anticipating various benefits of stability, Breiman (1996a,b) proposed bagging as an ensemble meta-algorithm to stabilize any base learning algorithm. Bagging, short for **b**ootstrap **agg**regat**ing**, refits the base algorithm to many perturbations of the training data and averages the resulting predictions. Breiman's vision of bagging as off-the-shelf stabilizer motivates our main question: *How stable is bagging on an arbitrary base algorithm, placing no assumptions on the data generating distribution?* In this paper, we first answer this

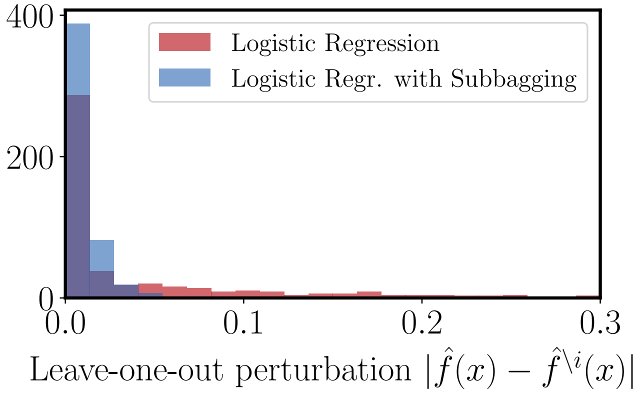

Figure 1: Distribution of leave-one-out perturbations for logistic regression (red) and subbagged logistic regression (blue), with $n = 500$ and $d = 200$. (See Section 6 for details on this simulation.)

question for the case of base algorithms with bounded outputs and then show extensions to the unbounded case.

1.1 Preview of Main Results

We study the following notion of algorithmic stability:

Definition 1 (Stability—informal version) *An algorithm is (ε, δ)-stable if, for any training data set $\mathcal{D}$ with n data points, and any test point x,*

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{\left|\hat{f}\left(x\right) - \hat{f}^{\setminus i}\left(x\right)\right| > \varepsilon\right\} \leq \delta, \tag{1}$$

where $\hat{f}$ is the model trained on the entire data set $\mathcal{D}$, while $\hat{f}^{\setminus i}$ is trained on the data set $\mathcal{D}$ with the ith data point removed.

In other words, this definition requires that, for *any* data set, if we drop one training point at random, then the resulting prediction produced by the algorithm is typically insensitive to this perturbation of the training data.

It is well known that, empirically, bagging and other ensembling procedures tend to improve the stability of an unstable base algorithm. For example, Figure 1 shows the histograms of leave-one-out perturbations $\left|\hat{f}\left(x\right) - \hat{f}^{\setminus i}\left(x\right)\right|$ for two different algorithms: logistic regression and logistic regression with subbagging (given by $\hat{f}_B\left(x\right) := \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{(b)}\left(x\right)$, where each $\hat{f}^{(b)}$ is a model fitted on $m = n/2$ out of n training data points sampled at random without replacement). We can clearly see that this perturbation is often far larger for logistic regression than for its subbagged version.

In this paper, we prove that stability (in the sense of Definition 1) is *automatically* achieved by the bagged version of any algorithm—with no assumptions on either the algorithm itself or on the training and test data, aside from requiring that the output predictions lie in a bounded range. A special case of our main result can be informally summarized as follows:

Theorem 2 (Main result—informal version) *Fix any algorithm with bounded output, and consider its subbagged version with m samples drawn without replacement,*

$$\hat{f}_B(x) := \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{(b)}(x),$$

where B is sufficiently large. Then the subbagged algorithm satisfies Definition 1 for any pair (ε,δ) satisfying

$$\delta\varepsilon^2 \gtrsim \frac{1}{n}\cdot\frac{p}{1-p} \tag{2}$$

where $p = \frac{m}{n}$.

(The formal version of Theorem 2, including many other forms of bagging, can be found in Section 4 below. We extend our main result to the unbounded case in Section 5.)

In the existing literature, relatively little is known about bagging's stabilization properties without additional assumptions on the base algorithm.[1] In this work, our stability guarantees (previewed in Theorem 2) will:

- apply to general base algorithms which may be highly unstable,
- hold for finite sample sizes,
- provide bounds that are optimal (up to constants), and
- hold deterministically, allowing for out-of-distribution test points and non-exchangeable data.

2. Algorithmic Stability

Consider a supervised learning setting with real responses. Formally, a learning algorithm $\mathcal{A}$ is a function that inputs a data set $\mathcal{D} = (Z_i)_{i=1}^n$ of pairs $Z_i = (X_i, Y_i)$ of covariates $X_i \in \mathcal{X}$ and responses $Y_i \in \mathcal{Y}$ and an auxiliary random variable $\xi \sim \mathrm{Unif}([0,1])$ and produces a fitted regression function $\hat{f} : \mathcal{X} \to \hat{\mathcal{Y}}$, given by $\hat{f} = \mathcal{A}\Big(\mathcal{D};\xi\Big)$. While many results in the literature consider only symmetric algorithms $\mathcal{A}$ (i.e., invariant to the ordering of the n training points in $\mathcal{D}$), here we do not constrain $\mathcal{A}$ to be symmetric.

The auxiliary random variable ξ may be viewed as a random seed, allowing for randomization in $\mathcal{A}$, if desired. For example, in many applications, we may wish to optimize an objective such as empirical risk, and then the resulting algorithm $\mathcal{A}$ consists of the specific numerical operations applied to the training data—for example, T steps of stochastic gradient descent (SGD) with a specific learning rate and batch size, with the random seed ξ

[1] We defer a more extensive discussion of prior work in this area to Section 7.3.

used for drawing the random batches in SGD. Our notation also allows for deterministic algorithms, since $\mathcal{A}$ is free to ignore the input argument ξ and depend only on the data.

There are many ways to define the stability of a learning algorithm. As noted by Shalev-Shwartz et al. (2010), every definition of stability quantifies the sensitivity of the output of $\mathcal{A}$ to small changes in the training set $\mathcal{D}$, but they all define 'sensitivity of the output' and 'small changes in the training set' differently. We present our main results for two definitions of stability and extend our results to many related notions in Section 5.3. One of the strongest possibilities is to require, for all data sets and all test points, that every prediction be insensitive to dropping any single observation. The following definition is closely related to *uniform prediction stability* (see, e.g., Dwork and Feldman, 2018).

Definition 3 *An algorithm $\mathcal{A}$ is worst-case (ε, δ)-stable if, for all data sets $\mathcal{D} = (Z_i)_{i=1}^n$ and test points $x \in \mathcal{X}$,*

$$\max_{i \in [n]} \mathbb{P}_\xi \left\{ \left| \hat{f}(x) - \hat{f}^{\setminus i}(x) \right| > \varepsilon \right\} \leq \delta, \tag{3}$$

where $\hat{f} = \mathcal{A}(\mathcal{D}; \xi)$, $\hat{f}^{\setminus i} = \mathcal{A}(\mathcal{D}^{\setminus i}; \xi)$ and $\mathcal{D}^{\setminus i} = (Z_j)_{j \neq i}$.

In many settings, however, this requirement is too stringent, since it forces $\mathcal{A}$ to be stable even when the most influential observation is dropped. A relaxation of this definition is the notion of average-case stability, where the perturbation comes from dropping one observation at random. Since we are primarily interested in average-case stability in this paper, we refer to it simply as '(ε, δ)-stable'.

Definition 4 *An algorithm $\mathcal{A}$ is (ε, δ)-stable if, for all data sets $\mathcal{D} = (Z_i)_{i=1}^n$ and test points $x \in \mathcal{X}$,*

$$\frac{1}{n} \sum_{i=1}^n \mathbb{P}_\xi \left\{ \left| \hat{f}(x) - \hat{f}^{\setminus i}(x) \right| > \varepsilon \right\} \leq \delta, \tag{4}$$

where $\hat{f} = \mathcal{A}(\mathcal{D}; \xi)$, $\hat{f}^{\setminus i} = \mathcal{A}(\mathcal{D}^{\setminus i}; \xi)$ and $\mathcal{D}^{\setminus i} = (Z_j)_{j \neq i}$.

This is the formal version of Definition 1, stated informally earlier—the difference here (aside from introducing notation) lies in the presence of the randomization term ξ.

The terminology '$\mathcal{A}$ *is* (ε, δ)*-stable*' in Definition 4 (or '$\mathcal{A}$ is worst-case (ε, δ)-stable, in Definition 3) suppresses the dependence on the sample size n. Since we are performing a non-asymptotic stability analysis, we treat n as a fixed positive integer throughout.

Clearly, Definition 4 is implied by Definition 3, but not vice versa; average-case stability thus relaxes worst-case stability to allow some small fraction of observations to have large leave-one-out perturbation. Average-case stability is a more permissive condition, and yet it is often sufficient for statistical inference. Indeed, if data points Z_i are exchangeable and $\mathcal{A}$ is symmetric, Condition (4) implies

$$\mathbb{P}_{\mathcal{D}, X_{n+1}, \xi} \left\{ |\hat{f}(X_{n+1}) - \hat{f}^{\setminus i}(X_{n+1})| > \varepsilon \right\} \leq \delta$$

for all i and for a new test point X_{n+1}, which is the condition of "out-of-sample stability" used by papers on distribution-free prediction mentioned in Section 1 (when $Z_{n+1} = (X_{n+1}, Y_{n+1})$

is a new test point that is exchangeable with the training data). Of course, our definition is a stronger property, as it is required to hold uniformly over any training set and any test point.

In fact, worst-case stability ensures an even stronger property—for a training sample $Z_1, \dots, Z_n$, it must hold that

$$\mathbb{P}_{\mathcal{D},\xi}\left\{|\hat{f}(X_i) - \hat{f}^{\setminus i}(X_i)| > \varepsilon\right\} \leq \delta$$

which is sometimes known as “in-sample stability”; informally, this bound implies that $\hat{f}$ is not “overfitted” to the training data, since its prediction $\hat{f}(X_i)$ at the ith training point is only slightly influenced by Z_i. On the other hand, average-case stability is not sufficient to ensure this type of bound, even on average over $i = 1, \dots, n$.

In many lines of the literature, it is more standard to define stability with respect to a loss function $\ell(\hat{f}\left(x\right), y)$. Define (ε, δ)-stability with respect to the loss ℓ as

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}_{\xi}\left\{|\ell(\hat{f}\left(x\right), y) - \ell(\hat{f}^{\setminus i}\left(x\right), y)| > \varepsilon\right\} \leq \delta.$$

If an algorithm $\mathcal{A}$ is (ε, δ)-stable in the sense of Definition 4, then it is $(\varepsilon/L, \delta)$-stable with respect to any loss function ℓ that is L-Lipschitz in its first argument. Hence, our stability guarantees immediately apply to any Lipschitz loss.[2]

Similarly, in the stability literature, it is more standard to control the expected value of the leave-one-out perturbation $\left|\hat{f}\left(x\right) - \hat{f}^{\setminus i}\left(x\right)\right|$ rather than controlling a tail probability. However, tail bounds can be easily converted to bounds in expectation using the standard identity

$$\mathbb{E}\left|\hat{f}\left(x\right) - \hat{f}^{\setminus i}\left(x\right)\right| = \int_0^\infty \mathbb{P}\left\{\left|\hat{f}\left(x\right) - \hat{f}^{\setminus i}\left(x\right)\right| > \varepsilon\right\} \mathrm{d}\varepsilon.$$

In Section 5.3, we define various related notions of stability more formally, and consider the implications of our main result for these alternative definitions of stability.

3. Bagging and its Variants

Bagging has a rich history in the machine learning literature (Breiman, 1996a; Dietterich, 2000; Valentini and Masulli, 2002) and is widely used in a variety of practical algorithms; random forests are a notable example (Breiman, 2001).

The theoretical properties of bagging have also been widely explored. For example, the stabilization properties of bagging have been studied for some specific base algorithms, such as trees (Basu et al., 2018) or k-means (Ben-David et al., 2007). Poggio et al. (2002) compared the stabilizing properties of bagging to those of ridge regression; LeJeune et al. (2020, 2024) recently established some deeper connections between the asymptotic risks of bagging and ridge regression (see also Patil et al. (2023)). Larsen (2023) showed bagging (where the base algorithm is empirical risk minimization) achieves optimal sample complexity for PAC learning in the realizable setting. Additional prior works have addressed the stability

[2] Assuming the loss ℓ is Lipschitz is standard in the literature—see, for example, Elisseeff et al. (2005); Hardt et al. (2016).

of bagging by proving that bagging, under certain conditions, increases the stability of an already stable algorithm (Elisseeff et al., 2005). In contrast, our results establish stability for bagging when applied to an arbitrary (and possibly highly unstable) base algorithm. We defer a more detailed discussion of prior work in this area to Section 7.3, where we can more fully compare to our own results.

Bagging applies resampling methods to reduce variance, smooth discontinuities, and induce stability in a base algorithm $\mathcal{A}$. The meta-algorithm repeatedly samples 'bags' from the training data $\mathcal{D}$, runs the base algorithm $\mathcal{A}$ on each bag, and averages the resulting models. Different resampling methods lead to some common variants:

- Classical bagging (Breiman, 1996a,b) samples m indices with replacement from $[n] = \{1, \ldots, n\}$.
- Subbagging (Andonova et al., 2002) samples m indices without replacement from $[n]$ (where $m \leq n$).

We distinguish 'classical bagging,' which employs sampling with replacement, from the more general 'bagging,' which we use to refer to any resampling method. For classical bagging, Breiman (1996a,b) originally proposed using the nonparametric bootstrap (Efron, 1979), that is, $m = n$, but $m \ll n$ is often computationally advantageous.

In both of the above strategies, because there are exactly m observations in each bag, there is a weak negative correlation between observations (that is, between the event that data point i is in the bag, and that data point j is in the bag). Randomizing the size of each bag is a standard trick that decorrelates these events:

- Poissonized bagging (Oza and Russell, 2001; Agarwal et al., 2014) samples M indices with replacement from $[n]$, where $M \sim \text{Poisson}(m)$.
- Bernoulli subbagging (Harrington, 2003) samples M indices without replacement from $[n]$, where $M \sim \text{Binomial}(n, \frac{m}{n})$.

Our stability results are quite flexible to the choice of resampling method, and in particular, apply to all four methods described above. In order to unify our results, we now present a generic version of bagging that includes all four of these variants as special cases.

3.1 Generic Bagging

Bagging is a procedure that converts any base algorithm $\mathcal{A}$ into a new algorithm, its bagged version $\widetilde{\mathcal{A}}_B$. Define

$$\text{seq}_{[n]} \coloneqq \{(i_1, \ldots, i_k) : k \geq 0, i_1, \ldots, i_k \in [n]\},$$

which is the set of finite sequences (of any length) consisting of indices in $[n]$. We refer to any $r \in \text{seq}_{[n]}$ as a "bag". Let $\mathcal{Q}_n$ denote a distribution on $\text{seq}_{[n]}$. For example, subbagging m out of n points corresponds to the uniform distribution over the set of length-m sequences $(i_1, \ldots, i_m)$ with distinct entries. Given a bag $r = (i_1, \ldots, i_m) \in \text{seq}_{[n]}$ and a data set $\mathcal{D} = (Z_1, \ldots, Z_n)$, define a new data set $\mathcal{D}_r = (Z_{i_1}, \ldots, Z_{i_m})$ selecting the data points according to the bag r.

Algorithm 1 Generic Bagging $\widetilde{\mathcal{A}}_B$

```
input Base algorithm $\mathcal{A}$; data set $\mathcal{D}$ with $n$ training points; number of bags $B \geq 1$;
  resampling distribution $\mathcal{Q}_n$
  for $b = 1, \ldots, B$ do
    Sample bag $r^{(b)} = (i_1^{(b)}, \ldots, i_{n_b}^{(b)}) \sim \mathcal{Q}_n$
    Sample seed $\xi^{(b)} \sim \text{Unif}([0, 1])$
    Fit model $\hat{f}^{(b)} = \mathcal{A}(\mathcal{D}_{r^{(b)}}; \xi^{(b)})$
  end for
output Averaged model $\hat{f}$ defined by
    $\hat{f}_B(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(x)$
```

To construct the bagged algorithm $\widetilde{\mathcal{A}}_B$ using $\mathcal{A}$ as a base algorithm, we first draw bags $r^{(1)}, \ldots, r^{(B)}$ from the resampling distribution $\mathcal{Q}_n$, then fit a model on each bag using $\mathcal{A}$, and average the resulting models for the final fitted function. Algorithm 1 summarizes this procedure.

Generic bagging treats the base algorithm $\mathcal{A}$ as a 'black-box,' in that it only accesses the base algorithm by querying it on different training sets and different random seeds. We write $\widetilde{\mathcal{A}}_B$ to denote the resulting algorithm obtained by applying generic bagging with $\mathcal{A}$ as the base algorithm.

Our theoretical analysis of bagging is simplified by considering an idealized version of generic bagging as the number of bags B tends to infinity. Our tactic is to directly study the stability of this large-B limit, and then derive analogous results for $\widetilde{\mathcal{A}}_B$ using simple concentration inequalities. To facilitate our theoretical analysis, we define in Algorithm 2 the limiting version of generic bagging.

Algorithm 2 Derandomized Bagging $\widetilde{\mathcal{A}}_\infty$

```
input Base algorithm $\mathcal{A}$; data set $\mathcal{D}$ with $n$ training points; resampling distribution $\mathcal{Q}_n$
output Averaged model $\hat{f}_\infty$ defined by
    $\hat{f}_\infty(x) = \mathbb{E}_{r,\xi}\left[\mathcal{A}(\mathcal{D}_r; \xi)(x)\right]$
  where the expectation is taken with respect to $r \sim \mathcal{Q}_n$ and $\xi \sim \text{Unif}([0, 1])$.
```

Note that the algorithm $\mathcal{A}$ may be a randomized algorithm, but derandomized bagging averages over any randomness in $\mathcal{A}$ (coming from the random seed ξ) as well as the randomness of the bags drawn from $\mathcal{Q}_n$.[3] For instance, the derandomized form of classical bagging averages uniformly over n^m possible subsets of the data; in practice, since we generally cannot afford n^m many calls to $\mathcal{A}$, we would instead run classical bagging with some large B as the number of randomly sampled bags.

[3] For $\widetilde{\mathcal{A}}_\infty$ to be defined, we assume that the expectation $\mathbb{E}_{r,\xi}\left[\mathcal{A}(\mathcal{D}_r; \xi)(x)\right]$ exists for all data sets $\mathcal{D}$ and test points x.

3.2 The Resampling Distribution $\mathcal{Q}_n$

To simplify the statement of the main results, we make a symmetry assumption on the resampling method $\mathcal{Q}_n$. All the variants we have described above (classical bagging, subbagging, Poissonized bagging, Bernoulli subbagging) satisfy this assumption.

Assumption 5 *The resampling method $\mathcal{Q}_n$ satisfies*

$$\mathcal{Q}_n\left\{(i_1,\dots,i_m)\right\} = \mathcal{Q}_n\left\{(\sigma(i_1),\dots,\sigma(i_m))\right\},$$

for all m, $i_1,\dots,i_m \in [n]$, and permutations $\sigma \in \mathcal{S}_n$.

Intuitively, this symmetry assumption requires the bagging algorithm to treat the indices $(1,\dots,n)$ as exchangeable (for example, bags $(1,2,2)$ and $(3,4,4)$ are equally likely).

Different bagging methods attain different degrees of stability. For instance, consider a degenerate case where $\mathcal{Q}_n$ returns a random permutation of $(1,\dots,n)$ (that is, subbagging with $m=n$). Then $\hat{f}_\infty$ is simply the result of running the base algorithm on shuffled versions of the data. In this case, the bagged algorithm is only as stable as the base algorithm. Our bounds on the stability of bagging depend on specific parameters of the resampling method $\mathcal{Q}_n$.

Definition 6 *For $\mathcal{Q}_n$ satisfying Assumption 5, let*

$$\begin{aligned} p &:= \mathbb{P}_{r\sim\mathcal{Q}_n}\left\{i \in r\right\}, \\ q &:= -\mathrm{Cov}_{r\sim\mathcal{Q}_n}(\mathbf{1}_{i\in r}, \mathbf{1}_{j\in r}), \end{aligned} \tag{5}$$

for any $i \neq j \in [n]$.

Here for a sequence $r=(i_1,\dots,i_m)\in \mathrm{seq}_{[n]}$, we write $i\in r$ to denote the event that $i_k = i$ for some k. Assumption 5 ensures that the value of p (and of q) are shared across all i (respectively, across all $i \neq j$).

We make the following restrictions on these parameters:

Assumption 7 *$\mathcal{Q}_n$ satisfies $p\in(0,1)$ and $q\geq 0$.*

The constraint $p\in(0,1)$ is a nondegeneracy assumption that guarantees a nonzero probability that any given observation Z_i gets excluded from some bags and included in others. The constraint $q \geq 0$ forces non-positive correlation between observations, that is, $i\in r$ does not increase the probability of $j \in r$ for $i\neq j$.

In our work, the role of the parameter q on our stability guarantees is always relatively insignificant. The symmetry condition imposed by Assumption 5 implies that the indicator variables $(\mathbf{1}_{i\in r})_{i\in[n]}$ are exchangeable. Since the covariance matrix of the random vector $(\mathbf{1}_{i\in r})_{i\in[n]}$ must be positive semidefinite, we always have the upper bound $q \leq \frac{p(1-p)}{n-1}$.

Table 1 provides values of p and q for the four different sampling schemes discussed above, which all satisfy Assumption 7. Classical bagging and subbagging both have a small positive q due to weak negative correlation between the events $i \in r$ and $j\in r$, while Poissonized bagging and Bernoulli subbagging decorrelate these events and so $q = 0$.

Algorithm	**Resampling method** $\mathcal{Q}_n$	$p = \mathbb{P}\{i \in r\}$	$q = -\mathrm{Cov}[i \in r, j \in r]$
Subbagging	$r = (i_1, \dots, i_m)$ drawn uniformly w/o replacement	$\frac{m}{n}$	$\frac{m(n-m)}{n^2(n-1)}$
Bernoulli subbagging	$M \sim \mathrm{Binomial}(n, \frac{m}{n})$ $r = (i_1, \dots, i_M)$ drawn uniformly w/o replacement	$\frac{m}{n}$	0
Classical bagging	$r \sim \mathrm{Unif}\{[n]^m\}$	$1 - \left(1 - \frac{1}{n}\right)^m$	$\left(1 - \frac{1}{n}\right)^{2m} - \left(1 - \frac{2}{n}\right)^m$
Poissonized bagging	$M \sim \mathrm{Poisson}(m)$ $r \mid M \sim \mathrm{Unif}\left\{[n]^M\right\}$	$1 - e^{-m/n}$	0

Table 1: Parameters p and q from Definition 6 for various sampling schemes $\mathcal{Q}_n$.

Since algorithmic stability compares a model fit on n observations to a model fit on $n-1$ observations, we need to specify resampling distributions at both sample sizes, that is, $\mathcal{Q}_n$ and $\mathcal{Q}_{n-1}$; naturally, to guarantee stability, these two distributions must be similar to each other. Specifically, we consider the setting where

$$\mathcal{Q}_{n-1} \text{ is given by the distribution of } r \sim \mathcal{Q}_n \text{ conditional on the event } r \in \mathrm{seq}_{[n-1]}, \tag{6}$$

that is, we are conditioning on the event that the nth data point is not contained in the bag. For example, if $\mathcal{Q}_n$ is chosen to be subbagging m out of n points (for some fixed $m \le n-1$), then $\mathcal{Q}_{n-1}$ is equal to the distribution obtained by subbagging m out of $n-1$ points.

4. Stability Guarantees for Bagging

In this section, we first present our main stability guarantee when the prediction range $\hat{\mathcal{Y}}$ is a bounded interval. We then show that this guarantee cannot be improved in general, up to a small multiplicative factor.

4.1 Main Result: Guarantee for Average-case Stability

We turn to our bound quantifying the average-case stability of derandomized bagging. In this section, we restrict our attention to settings where the output regression function $\hat{f}$ is bounded. We consider the unbounded case in Section 5.1.

To examine the stability of $\widetilde{\mathcal{A}}_\infty$ (obtained by applying derandomized bagging to a base algorithm $\mathcal{A}$) our stability results compare the models:

- $\hat{f}_\infty$, obtained by running derandomized bagging (Algorithm 2) with base algorithm $\mathcal{A}$, data set $\mathcal{D}$, and sampling distribution $\mathcal{Q}_n$; and
- $\hat{f}_\infty^{\setminus i}$, obtained by running derandomized bagging (Algorithm 2) with base algorithm $\mathcal{A}$, data set $\mathcal{D}^{\setminus i}$, and sampling distribution $\mathcal{Q}_{n-1}$, constructed as in (6).

Theorem 8 *Let $\hat{\mathcal{Y}} = [0,1]$.*[4] *Fix a distribution $\mathcal{Q}_n$ on $\mathrm{seq}_{[n]}$ satisfying Assumptions 5 and 7, and let $\mathcal{Q}_{n-1}$ be defined as in (6). For any algorithm $\mathcal{A}$, derandomized bagging $\widetilde{\mathcal{A}}_\infty$ is (ε,δ)-stable provided*

$$\delta\varepsilon^2 \geq \frac{1}{4n}\left(\frac{p}{1-p} + \frac{q}{(1-p)^2}\right). \tag{7}$$

In particular, since $q \leq \frac{p(1-p)}{n-1}$, the above bound implies that (ε,δ)-stability holds as long as

$$\delta\varepsilon^2 \geq \frac{1}{4(n-1)} \cdot \frac{p}{1-p}. \tag{8}$$

We prove Theorem 8, along with all subsequent results, in Appendix A.[5]

A simple application of Hoeffding's inequality leads to a similar stability guarantee for generic bagging. In this result, we compare the models:

- $\hat{f}_B$, obtained by running generic bagging (Algorithm 1) with base algorithm $\mathcal{A}$, data set $\mathcal{D}$, and resampling distribution $\mathcal{Q}_n$; and
- $\hat{f}_B^{\setminus i}$, obtained by running generic bagging (Algorithm 1) with base algorithm $\mathcal{A}$, data set $\mathcal{D}^{\setminus i}$, and resampling distribution $\mathcal{Q}_{n-1}$, constructed as in (6).

Theorem 9 *Let $\hat{\mathcal{Y}} = [0,1]$. Fix a distribution $\mathcal{Q}_n$ on $\mathrm{seq}_{[n]}$ satisfying Assumptions 5 and 7, and let $\mathcal{Q}_{n-1}$ be defined as in (6). For any algorithm $\mathcal{A}$ and any $B \geq 1$, generic bagging $\widetilde{\mathcal{A}}_B$ is $\left(\varepsilon + \sqrt{\frac{2}{B}\log\left(\frac{4}{\delta'}\right)}, \delta + \delta'\right)$-stable for any (ε,δ) satisfying Condition (7) and any $\delta' > 0$.*

If derandomized bagging is guaranteed to satisfy (ε,δ) stability via (7), then we may take $B \geq \frac{2}{\varepsilon^2}\log\left(\frac{4}{\delta}\right)$ to guarantee $(2\varepsilon, 2\delta)$-stability of generic bagging. For instance, if $p \in (0,1)$ and $\delta \in (0,1)$ are regarded as constants, Theorem 8 guarantees stability of derandomized bagging as long as $\varepsilon \gtrsim \frac{1}{\sqrt{n}}$. In order to guarantee the same level of stability for generic bagging, we need the number of bags B to be of the same order as the number of observations n, which is typically unrealistic in practice. More generally, for any fixed B, the result accounts for the Monte Carlo error in the generic bagging algorithm.

4.1.1 Sampling Regimes for Subbagging

Theorem 8 covers a wide range of regimes depending on the choices of ε, δ and p. In this section, we give some concrete examples in the case of subbagging, to build intuition:

Proportional subsampling with $m \propto n$: Suppose we employ subbagging with $m = n/2$. The stability condition (7) in the theorem simplifies to $\delta\varepsilon^2 \geq \frac{1}{4(n-1)}$. More generally, for

[4] All theoretical results in this section are stated for $\hat{\mathcal{Y}} = [0,1]$ for simplicity, but it is straightforward to generalize to the case $\hat{\mathcal{Y}} = [a,b]$ by simply replacing (ε,δ)-stability with $(\varepsilon(b-a),\delta)$-stability in the guarantee.

[5] The proof shows a slightly stronger notion of stability, where $\mathbb{P}_\xi\left\{\left|\hat{f}(x) - \hat{f}^{\setminus i}(x)\right| > \varepsilon\right\}$ in Equation (4) is replaced with $\mathbb{P}_\xi\left\{\left|\hat{f}(x) - \hat{f}^{\setminus i}(x)\right| \geq \varepsilon\right\}$.

$m = O(n)$, stability holds with $\delta\varepsilon^2 \gtrsim \frac{1}{n}$. Hence, our stability result applies in a variety of regimes. For instance, if $\delta > 0$ does not depend on n, bagging satisfies average-case (ε, δ)-stability with $\varepsilon = O(n^{-1/2})$. We may also take $\varepsilon > 0$ fixed and $\delta = O(n^{-1})$, or even $\delta = \varepsilon = O(n^{-1/3})$ going to zero simultaneously.

Massive subsampling with $m = o(n)$: For massive data sets, it may be computationally advantageous to subsample a very small fraction of the data (Kleiner et al., 2014). *Massive subsampling*, where we take bags of size $m = O(n^\kappa)$ for some $\kappa \in (0, 1)$, can be seen to further enhance stability via our result above. In this case, condition (7) becomes $\delta\varepsilon^2 \gtrsim \frac{1}{n^{2-\kappa}}$. See Section 7.3 for a discussion of results in the literature in this regime.

Minimal subsampling with $m = n - o(n)$: Massive subsampling, or even subsampling a constant fraction of the data, often comes with some loss of statistical efficiency. To avoid this, our result even allows for resampling schemes with $m = n - o(n)$, that is, each subsample contains nearly the entire data set. For example, taking $m = n - n^\kappa$ for some $\kappa \in (0, 1)$, condition (7) becomes $\delta\varepsilon^2 \gtrsim \frac{1}{n^\kappa}$.

4.2 Tightness of Stability Guarantee

In the special case of subbagging, we show that Theorem 8 cannot be improved (beyond a constant factor) without assuming more about the base algorithm. We only state this result in the ideal, derandomized case, since this is typically more stable than its finite B counterpart.

Theorem 10 *Let $\hat{\mathcal{Y}} = [0, 1]$. Fix $n > m \geq 1$ and $\delta \in (0, 1/2)$. There is a base algorithm $\mathcal{A}^\sharp$ such that subbagging $\widetilde{\mathcal{A}}^\sharp_\infty$ with m out of n observations is not (ε, δ)-stable for any*

$$\varepsilon < \left(1 - \delta - n^{-1}\right) p \, \mathbb{P}\left\{H = \Big\lfloor p\left(1 + \lfloor n\delta \rfloor\right) \Big\rfloor\right\}, \tag{9}$$

where $p = \frac{m}{n}$ and where the probability is taken with respect to $H \sim \mathrm{HyperGeometric}(n - 1, \lfloor n\delta \rfloor, m)$.

To see how this result compares to the guarantee given in Theorem 8, consider a simple case where $n\delta$ and $m\delta = np\delta$ are integers, and take $p < 1$. Then

$$\mathbb{P}\left\{H = \Big\lfloor p\left(1 + \lfloor n\delta \rfloor\right) \Big\rfloor\right\} = \mathbb{P}\left\{H = np\delta\right\} = \frac{\binom{n\delta}{np\delta} \cdot \binom{n(1-\delta)-1}{np(1-\delta)}}{\binom{n-1}{np}} \approx \frac{1}{\sqrt{2\pi n\delta(1-\delta)p(1-p)}},$$

where the last step holds by taking Stirling's approximation to each factorial term in each Binomial coefficient (and the approximation is accurate as long as $n \cdot \min\{\delta, 1-\delta\} \cdot \min\{p, 1-p\}$ is large). Thus the right-hand side of (9) is approximately

$$\approx \frac{1}{\sqrt{2\pi n}} \cdot \sqrt{\frac{1-\delta}{\delta} \cdot \frac{p}{1-p}}.$$

Since we have assumed $\delta < 1/2$, we therefore see that stability fails for $\mathcal{A}^\sharp$ when (approximately)

$$\delta\varepsilon^2 < \frac{1}{4\pi n} \cdot \frac{p}{1-p}.$$

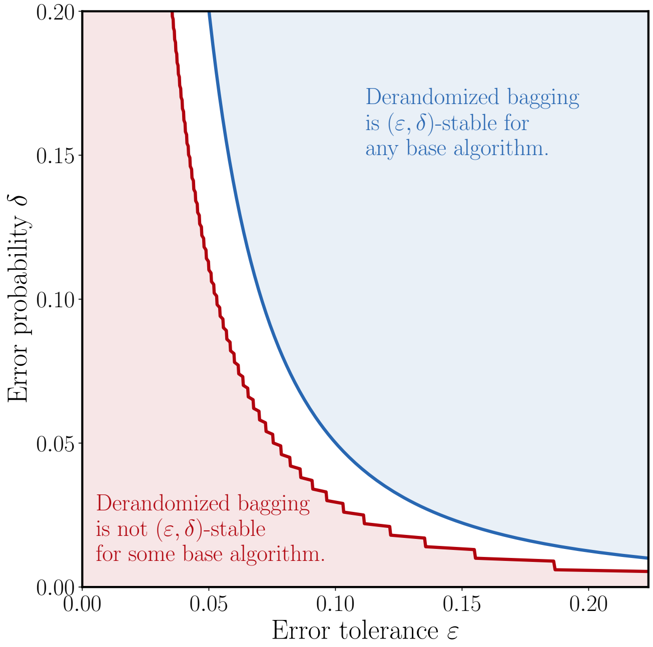

Figure 2: Phase diagram comparing Theorems 8 and 10, with $n = 500, p = 0.5$.

Up to a constant, this matches the leading term of the stability guarantee in Theorem 8, demonstrating the tightness of our guarantee.

In Figure 2, we plot a phase diagram comparing the stability guarantee (7) with the tightness condition (9) for finite n. We take $n = 500, p = 1/2$, and $q = p(1-p)/(n-1) = 1/1996$, which are the values of p and q for subbagging with $m = n/2$ (see Table 1). The blue line shows, for each δ, the minimum ε satisfying (7), and the shaded blue region shows additional (ε, δ) pairs satisfying the inequality. This means that, for any base algorithm $\mathcal{A}$ with outputs in $\hat{\mathcal{Y}} = [0,1]$, its subbagged version is guaranteed to satisfy (ε,δ)-stability for any pair (ε,δ) in the blue shaded region. Similarly, the red line shows, for each δ, the maximum ε satisfying (9). This means that, for any (ε,δ) in the red shaded region, we can construct an algorithm $\mathcal{A}^\sharp$, again with outputs in $\hat{\mathcal{Y}} = [0,1]$, such that its subbagged version fails to be (ε,δ)-stable. The narrow white region between the two conditions illustrates the small gap between the two results.

5. Extensions

In this section, we consider various extensions of our main result. We first discuss two approaches to the case of unbounded outputs. Next, we show a hardness result explaining why we cannot obtain a similar guarantee for worst-case stability. Finally, we consider the implications of our main result for various alternative definitions of stability.

5.1 Unbounded Outputs

We next extend our main result to algorithms $\mathcal{A}$ with unbounded output $\hat{\mathcal{Y}} = \mathbb{R}$. For derandomized bagging $\widetilde{\mathcal{A}}_\infty$ to be well-defined, we assume that the expectation

$$\mathbb{E}_{r,\xi}\left[\mathcal{A}(\mathcal{D}_r;\xi)(x)\right]$$

exists. For instance, for classical bagging, subbagging, and Bernoulli subbagging, the average over $r \sim \mathcal{Q}_n$ constitutes a finite sum, so we are simply assuming that expectation over the random seed

$$\mathbb{E}_{\xi}\left[\mathcal{A}(\mathcal{D};\xi)(x)\right]$$

exists for any fixed data set $\mathcal{D}$.

In order to establish some control over the scale of the outputs of the fitted model, we extend our definition of average-case stability to allow for a data-dependent component. Consider for instance any algorithm $\mathcal{A}$ with $\hat{\mathcal{Y}} = [0,1]$, and define a new algorithm $\mathcal{A}'$ scaling the outputs by $R > 0$, that is, $\mathcal{A}'(\mathcal{D};\xi) = R \cdot \mathcal{A}(\mathcal{D};\xi)$. If the original algorithm $\mathcal{A}$ is (ε,δ)-stable, then the scaled algorithm $\mathcal{A}'$ is $(\varepsilon R,\delta)$-stable.

We might hope that we can take R to be the empirical range of the algorithm,

$$R = \mathrm{Range}(\mathcal{D},x) = \sup_{r:\mathcal{Q}_n(\{r\})>0} \mathbb{E}_{\xi}\left[\mathcal{A}(\mathcal{D}_r;\xi)(x)\right] - \inf_{r:\mathcal{Q}_n(\{r\})>0} \mathbb{E}_{\xi}\left[\mathcal{A}(\mathcal{D}_r;\xi)(x)\right].$$

However, since this quantity depends (in general) on $\mathcal{D}$ and on x, it would not be well-defined to claim that $\mathcal{A}$ is $(\varepsilon R,\delta)$-stable universally across all $\mathcal{D}$ and all x.

Instead, to allow for a data-dependent range, we consider scaling ε by a data-dependent scale parameter $\mathcal{R}(\mathcal{D},x)$, where

$$\mathcal{R} : \bigcup_{n\geq 0} (\mathcal{X}\times\mathcal{Y})^n \times \mathcal{X} \to \mathbb{R}_+.$$

We now define $(\varepsilon,\delta,\mathcal{R})$-stability to account for data-dependent changes in scale.

Definition 11 *Let $\varepsilon,\delta \geq 0$ and let $\mathcal{R}$ denote a data-dependent range (formally defined above). An algorithm $\mathcal{A}$ is $(\varepsilon,\delta,\mathcal{R})$-stable if, for all data sets $\mathcal{D} = (Z_i)_{i=1}^n$ of size n and all test points $x \in \mathcal{X}$,*

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}_{\xi}\left\{\left|\hat{f}(x) - \hat{f}^{\setminus i}(x)\right| > \varepsilon\,\mathcal{R}(\mathcal{D},x)\right\} \leq \delta, \tag{10}$$

where $\hat{f} = \mathcal{A}(\mathcal{D};\xi)$, $\hat{f}^{\setminus i} = \mathcal{A}(\mathcal{D}^{\setminus i};\xi)$ and $\mathcal{D}^{\setminus i} = (Z_j)_{j\neq i}$.

Inspecting the proof of Theorem 8, we only use boundedness to control the variance of our model predictions $\mathbb{E}_{\xi}\left[\mathcal{A}(\mathcal{D}_r;\xi)(x)\right]$ as a function of the random bag $r \sim \mathcal{Q}_n$. This observation leads to the following, more general result.

Theorem 12 *Let $\hat{\mathcal{Y}} = \mathbb{R}$. Fix a distribution $\mathcal{Q}_n$ on $\mathrm{seq}_{[n]}$ satisfying Assumptions 5 and 7, and let $\mathcal{Q}_{n-1}$ be defined as in* (6)*. Let (ε,δ) satisfy Equation* (7)*. For any algorithm $\mathcal{A}$, derandomized bagging $\widetilde{\mathcal{A}}_\infty$ is $(\varepsilon,\delta,\mathcal{R}^*)$-stable, where*

$$\mathcal{R}^*(\mathcal{D},x) := 2\sqrt{\mathrm{Var}_{r\sim\mathcal{Q}_n}\left(\mathbb{E}_{\xi}\left[\mathcal{A}(\mathcal{D}_r;\xi)(x)\right]\right)} \leq \mathrm{Range}(\mathcal{D},x). \tag{11}$$

As long as $\mathbb{E}_\xi\left[\mathcal{A}(\mathcal{D}_r;\xi)(x)\right]$ is well-defined for every r, the range $\mathrm{Range}(\mathcal{D},x)$ is automatically finite for any $\mathcal{Q}_n$ with finite support. Furthermore, Theorem 12 strictly generalizes the stability guarantee of Theorem 8, since $\mathcal{R}^*(\mathcal{D},x) \leq \mathrm{Range}(\mathcal{D},x) \leq 1$ in the case $\hat{\mathcal{Y}} = [0,1]$. We present a weaker result for the finite-B regime in Appendix C.

5.1.1 Alternative Approach: Adaptive Clipping

In some settings, for example, with heavy tailed responses, the range in the previous display or the standard deviation in Equation (11) may be prohibitively large. One way to reduce the standard deviation $\mathcal{R}^*(\mathcal{D},x)$ is to post-process the algorithm $\mathcal{A}$. We next consider the advantages of clipping the output of $\mathcal{A}$ to secure greater stability.

Given an interval $I = [l,u]$ and a response $\hat{y} \in \mathbb{R}$, the clipped response $\mathrm{Clip}_I(\hat{y})$ is defined as

$$\mathrm{Clip}_I(\hat{y}) \coloneqq \max\left\{l, \min\left\{\hat{y}, u\right\}\right\}. \tag{12}$$

In Algorithm 3, we define a variant of the derandomized bagging algorithm that allows the individual bagged predictions $\hat{f}_\infty^{(r)}(x)$ to be clipped to some interval $I = I(\mathcal{D})$ that depends on the full data set. We write $\widetilde{\mathcal{A}}_{B,I}$ to denote the algorithm obtained by applying adaptively clipped bagging with $\mathcal{A}$ as the base algorithm. Stability of Algorithm 3 does not follow immediately from Theorem 12 because $I(\mathcal{D})$ may not be the same as $I(\mathcal{D}^{\setminus i})$, so the algorithm being bagged is itself changing when we perturb the training data. For simplicity, we state our result for the derandomized limit $\widetilde{\mathcal{A}}_{\infty,I}$ and give a finite B version in Appendix C.

Algorithm 3 Adaptively Clipped Bagging

```
input Base algorithm $\mathcal{A}$; data set $\mathcal{D}$ with $n$ training points; number of bags $B \geq 1$;
  resampling distribution $\mathcal{Q}_n$; data-dependent range $I(\cdot)$
  for $b = 1, \dots, B$ do
    Sample bag $r^{(b)} = (i_1^{(b)}, \dots, i_{n_b}^{(b)}) \sim \mathcal{Q}_n$
    Sample seed $\xi^{(b)} \sim \mathrm{Unif}([0,1])$
    Fit model $\hat{f}^{(b)} = \mathcal{A}(\mathcal{D}_{r^{(b)}}; \xi^{(b)})$
  end for
output Averaged model $\hat{f}_{B,I}$ defined by
    $\hat{f}_{B,I}(x) = \frac{1}{B}\sum_{b=1}^B \mathrm{Clip}_{I(\mathcal{D})}(\hat{f}^{(b)}(x))$
```

Theorem 13 *Let $\hat{\mathcal{Y}} = \mathbb{R}$. Fix a distribution $\mathcal{Q}_n$ on $\mathrm{seq}_{[n]}$ satisfying Assumptions 5 and 7, and let $\mathcal{Q}_{n-1}$ be defined as in* (6). *Suppose the mapping $I(\cdot)$ from data sets to intervals satisfies*

$$\frac{1}{n}\sum_{i=1}^n \mathbf{1}\left\{I\left(\mathcal{D}\right) \neq I\left(\mathcal{D}^{\setminus i}\right)\right\} \leq \delta_I. \tag{13}$$

Let $\mathcal{R}(\mathcal{D},x) = \mathrm{length}(I(\mathcal{D}))$ and let (ε,δ) satisfy Equation (7). *For any algorithm $\mathcal{A}$, derandomized adaptively clipped bagging $\widetilde{\mathcal{A}}_{\infty,I}$ is $(\varepsilon, \delta+\delta_I, \mathcal{R})$-stable.*

As a special case, consider taking $I(\mathcal{D})$ to be the observed range, that is,

$$I(\mathcal{D}) = \left[\min_i Y_i, \max_i Y_i\right].$$

When $(Z_i)_{i=1}^{n+1}$ are exchangeable random variables, we can apply Theorem 13 with $\delta' = \frac{2}{n}$. Restricting to the empirical range of the Y_i's does not substantially limit the learned regression function $\hat{f}_{\infty,I}$—it simply requires that predictions cannot lie outside the observed range of the training data (which is already satisfied by many base algorithms, such as nearest neighbors or regression trees, and typically would not substantially alter the output of many other algorithms). More generally, we can take $I(\mathcal{D}) = [Y_{(k)}, Y_{(n+1-k)}]$ for some fixed $k < n/2$, where $Y_{(1)} \leq \cdots \leq Y_{(n)}$ denote the order statistics of $Y_1, \ldots, Y_n$. In this case, we have $\delta' = \frac{2k}{n}$ in (13), which allows for some fraction δ' of outliers to be removed when constructing the data-dependent range, thus ensuring that $\mathcal{R}(\mathcal{D}, x)$ is not too large.

Of course, there are many other potential strategies for defining the data-dependent range $I(\mathcal{D})$, and the benefits and drawbacks of these various choices depend on the specific data distribution and base algorithm. Exploring these options, and designing practical versions of this procedure to provide accurate fitted models with meaningful stability guarantees, is an important question for future work.

5.2 Hardness of Worst-case Stability

Our results above establish that (ε, δ)-stability can be guaranteed for any (bounded) base algorithm even for very small ε—for instance, taking $p \in (0, 1)$ to be a constant, we can choose $\varepsilon = O(n^{-1/2})$. Next, we show that no analogous result exists for worst-case stability—indeed, for this stricter definition, stability cannot be guaranteed for any $\varepsilon < p$, and therefore $\varepsilon = O(n^{-1/2})$ can only be guaranteed via massively subsampling the data with $p = O(n^{-1/2})$.

Theorem 14 *Fix $\mathcal{Q}_n$ and let $\hat{\mathcal{Y}} = [0, 1]$.*

(i) For any algorithm $\mathcal{A}$, derandomized bagging is worst-case (p, δ)-stable for all δ.

(ii) If $|\mathcal{X}| > 1$, there is a base algorithm $\mathcal{A}^\dagger$ such that derandomized bagging is not worst-case (ε, δ)-stable for any $\varepsilon < p$ and $\delta < 1$.

Part *(i)* of the theorem has repeatedly appeared in various forms (see, e.g., Poggio et al. (2002, Theorem 3.1), Elisseeff et al. (2005, Proposition 4.3) and Chen et al. (2022, Theorem 5)); in Section 7.3 we discuss how this observation has led some authors on algorithmic stability to advocate for subsampling a decreasing fraction of the data $m = o(n)$ as $n \to \infty$. In contrast, by moving to average-case stability, our results allow $m = O(n)$ and even $m = n - o(n)$, enabling far greater accuracy in the fitted models.

The base algorithm $\mathcal{A}^\dagger$ in the proof of part *(ii)* of the theorem memorizes the training data:

$$\mathcal{A}^\dagger(\mathcal{D})\,(x) := \mathbf{1}\left\{\exists\, (\tilde{x}, \tilde{y}) \in \mathcal{D} : \tilde{x} = x\right\}.$$

If $x = x_i$ for precisely one $i \in [n]$, then this training point (x_i, y_i) has maximal influence on the value of $\hat{f}_\infty\,(x)$—every bag containing (x_i, y_i) predicts 1, and every bag not containing (x_i, y_i) leads to a predicts 0. This counterexample can be used to show an even stronger hardness result, for average-case, "in-sample" stability (discussed earlier in Section 2): if $x_1, \ldots, x_n$ are all distinct,

$$\frac{1}{n}\sum_{i=1}^{n} 1\left\{|\hat{f}^{\dagger}_{\infty}(x_i) - \hat{f}^{\dagger\setminus i}_{\infty}(x_i)| > \varepsilon\right\} = \begin{cases} 1 & \text{if } \varepsilon \leq p \\ 0 & \text{if } \varepsilon > p \end{cases},$$

where $\hat{f}^{\dagger}_{\infty} = \mathcal{A}^{\dagger}_{\infty}(\mathcal{D})$. Note, however, that for a fixed x, *at most one* index $i \in [n]$ can change the bagged prediction by p. This limitation of $\mathcal{A}^{\dagger}$ provides useful intuition for why we may expect a stronger result for our main definition of (ε, δ)-stability, Definition 4, where the test point $x \in \mathcal{X}$ is fixed.

5.3 Alternative Frameworks for the Main Result

In this section, we discuss various implications of our main stability guarantee for related criteria.

5.3.1 Stability in Expectation

In Definition 4, average-case algorithmic stability controls the tail of the distribution of leave-one-out perturbations. Some authors (e.g., Bousquet and Elisseeff, 2002; Elisseeff et al., 2005) prefer to work with the expected value of the leave-one-out perturbation. We can consider a version of average-case stability that works with expected values rather than probabilities, requiring that

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\xi}\left|\hat{f}\,(x) - \hat{f}^{\setminus i}\,(x)\right| \leq \beta, \tag{14}$$

for all data sets $\mathcal{D}$ of size n and test points $x \in \mathcal{X}$, where $\hat{f} = \mathcal{A}(\mathcal{D};\xi)$ and $\hat{f}^{\setminus i} = \mathcal{A}(\mathcal{D}^{\setminus i};\xi)$.

Corollary 15 *In the setting of Theorem 8, for any B, generic bagging $\widetilde{\mathcal{A}}_B$ satisfies stability condition* (14) *at level $\beta = \beta_{n,m,B}$, where*

$$\beta_{n,m,B} = \sqrt{\frac{1}{4n}\left(\frac{p}{1-p} + \frac{q}{(1-p)^2}\right)} + \sqrt{\frac{2\pi}{B}}. \tag{15}$$

Note that the scaling of β in this result is comparable to the scaling of ε in Theorem 9 if we take δ, δ' to be constant. The result holds for any B, including the case of derandomized bagging by taking $B \to \infty$.

5.3.2 Stability in the Loss

Building on earlier definitions of stability (Kearns and Ron, 1999; Bousquet and Elisseeff, 2002), Elisseeff et al. (2005, Definition 7) say that a randomized algorithm $\mathcal{A}$ satisfies *random hypothesis stability* at level β with respect to the loss function ℓ and distribution P if the following holds:

$$\forall i \in \{1, \ldots, n\},\ \mathbb{E}_{(X_i,Y_i)_{i=1}^{n+1} \overset{\text{iid}}{\sim} P,\xi}\left|\ell(\hat{f}(X_{n+1}), Y_{n+1}) - \ell(\hat{f}^{\setminus i}(X_{n+1}), Y_{n+1})\right| \leq \beta, \tag{16}$$

where $\hat{f} = \mathcal{A}(\mathcal{D}; \xi)$, $\hat{f}^{\setminus i} = \mathcal{A}(\mathcal{D}^{\setminus i}; \xi)$, $\mathcal{D} = (X_i, Y_i)_{i=1}^n$. Our next result records the straightforward observation that Corollary 15 implies random hypothesis stability with respect to any loss ℓ that is Lipschitz in its first argument.

Corollary 16 *Let $\hat{\mathcal{Y}} = [0, 1]$. Fix a distribution $\mathcal{Q}_n$ on $\mathrm{seq}_{[n]}$ satisfying Assumptions 5 and 7, and let $\mathcal{Q}_{n-1}$ be defined as in* (6)*. Let P denote any distribution on $\mathcal{X} \times \mathcal{Y}$, and let $\ell : \hat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}_+$ denote any loss function that is L-Lipschitz in its first argument. For any algorithm $\mathcal{A}$ and any B, generic bagging $\widetilde{\mathcal{A}}_B$ satisfies random hypothesis stability* (16) *at level $\beta = L\beta_{n,m,B}$, where $\beta_{n,m,B}$ is defined as in Equation* (15)*.*

In fact, our main result implies that the inequality in (16) holds (on average over $i \in [n]$) even *conditional on the training data $\mathcal{D}$ and the test point* (X_{n+1}, Y_{n+1}), eliminating the assumption that the data are iid—in fact, in our result, the test point can be adversarially chosen.

5.3.3 Replace-one Stability

The stability definitions in this paper concern the leave-one-out perturbation $\left|\hat{f}(x) - \hat{f}^{\setminus i}(x)\right|$. Alternative definitions, used for example by Shalev-Shwartz et al. (2010), are obtained by considering a 'replace-one' perturbation $|\hat{f}(x) - \hat{f}^{(i)}(x)|$, where

$$\hat{f}^{(i)} = \mathcal{A}(\mathcal{D}^{(i)}; \xi) \qquad \text{and} \qquad \mathcal{D}^{(i)} = \mathcal{D}^{\setminus i} \cup (Z_i').$$

We say that a randomized algorithm $\mathcal{A}$ satisfies *random replace-one hypothesis stability β* with respect to the loss function ℓ and distribution P if the following holds:

$$\forall i \in \{1, \ldots, n\}, \ \mathbb{E}_{(X_i,Y_i)_{i=1}^n, (X_i',Y_i') \overset{\text{iid}}{\sim} P, \xi} \left|\ell(\hat{f}(x), y) - \ell(\hat{f}^{(i)}(x), y)\right| \le \beta. \tag{17}$$

Stability to leave-one-out perturbations is typically stronger than stability to replace-one perturbations. To see this, note that, by the triangle inequality, the replace-one perturbation can be bounded as

$$|\hat{f}(x) - \hat{f}^{(i)}(x)| \le \left|\hat{f}(x) - \hat{f}^{\setminus i}(x)\right| + |\hat{f}^{(i)}(x) - \hat{f}^{\setminus i}(x)|,$$

where both terms on the right-hand-side are leave-one-out perturbations. A guarantee for replace-one stability thus follows immediately from Corollary 15.

Corollary 17 *In the setting of Corollary 16, generic bagging satisfies random replace-one hypothesis stability at level $\beta = 2L\beta_{n,m,B}$, where $\beta_{n,m,B}$ is defined in Equation* (15) *and $\hat{f}_B^{(i)}$ is obtained by running generic bagging (Algorithm 1) with base algorithm $\mathcal{A}$, data set $\mathcal{D}^{(i)}$, and resampling distribution $\mathcal{Q}_n$.*

6. Experiments

In this section, we study the stability of subbagging in simulation experiments. We use scikit-learn (Pedregosa et al., 2011) for all base algorithms. Code to reproduce all experiments is available at `https://github.com/jake-soloff/subbagging-experiments`.

6.1 Data and Methods

We consider four simulation settings:

- **Setting 1:** We simulate from the following data generating process:

$$X_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d),\ Y_i \mid X_i \overset{\text{ind}}{\sim} \text{Bernoulli}\left(\frac{1}{1+\exp\left(-X_i^\top \theta^*\right)}\right),$$

  with sample size $n = 500$ and dimension $d = 200$, and where $\theta^* = (.1, \dots, .1) \in \mathbb{R}^d$. The base algorithm $\mathcal{A}$ is the output of ℓ_2-regularized logistic regression, given by $\mathcal{A}(\mathcal{D})\,(x) = \hat{f}_{\hat{\theta}}\,(x) := \left(1 + e^{-x^\top \hat{\theta}}\right)^{-1}$, where

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \left\{ C \sum_{i=1}^n \left(-Y_i \log(\hat{f}_\theta(X_i)) - (1 - Y_i)\log(1 - \hat{f}_\theta(X_i))\right) + \frac{1}{2}\|\theta\|_2^2 \right\}.$$

  We use `sklearn.linear_model.LogisticRegression`, setting options `penalty='l2'`, `C=1e3/n` and `fit_intercept=False`, leaving all other parameters at their default values.

- **Setting 2:** Same as Setting 1, changing only the sample size to $n = 1000$.

- **Setting 3:** Same as Setting 1, changing only the base algorithm $\mathcal{A}$ to a neural network with a single hidden layer. We use `sklearn.neural_network.MLPClassifier`, setting `hidden_layer_sizes=(40,)`, `solver="sgd"`, `learning_rate_init=0.2`, `max_iter=8`, and `alpha=1e-4`, leaving all other parameters at their default values.

- **Setting 4:** We simulate from the following data generating process:

$$\begin{aligned}(X_i, \alpha_i, \gamma_i) &\overset{\text{iid}}{\sim} \text{Unif}([0,1]^d) \times \text{Unif}([-.25, .25]) \times \text{Unif}([0,1]),\\ Y_i &= \sum_{j=1}^d \sin\left(\frac{X_{ij}}{j}\right) + \alpha_i \mathbf{1}\left\{i = 1 \pmod 3\right\} + \gamma_i \mathbf{1}\left\{i = 1 \pmod 4\right\},\end{aligned}$$

  with $n = 500$ and $d = 40$. Note that the algorithm has access to the observed data $\mathcal{D} = (X_i, Y_i)_{i=1}^n$, that is, α_i and γ_i are latent variables used only to generate the data $\mathcal{D}$. We apply `sklearn.tree.DecisionTreeRegressor` to train the regression trees, setting `max_depth=50` and leaving all other parameters at their default values.

6.2 Results

Our results are shown in Figure 3. In each setting, we apply the base algorithm $\mathcal{A}$ as well as subbagging $\widetilde{\mathcal{A}}_B$ with $m = n/2$ samples in each bag, using $B = 10000$ bags. The left panels of Figure 3 show the histogram of leave-one-out perturbations $\left|\hat{f}\,(x) - \hat{f}^{\setminus i}\,(x)\right|$ for $i \in \{1, \dots, n\}$. In the right panels of Figure 3, for a fixed data set and algorithm, we measure

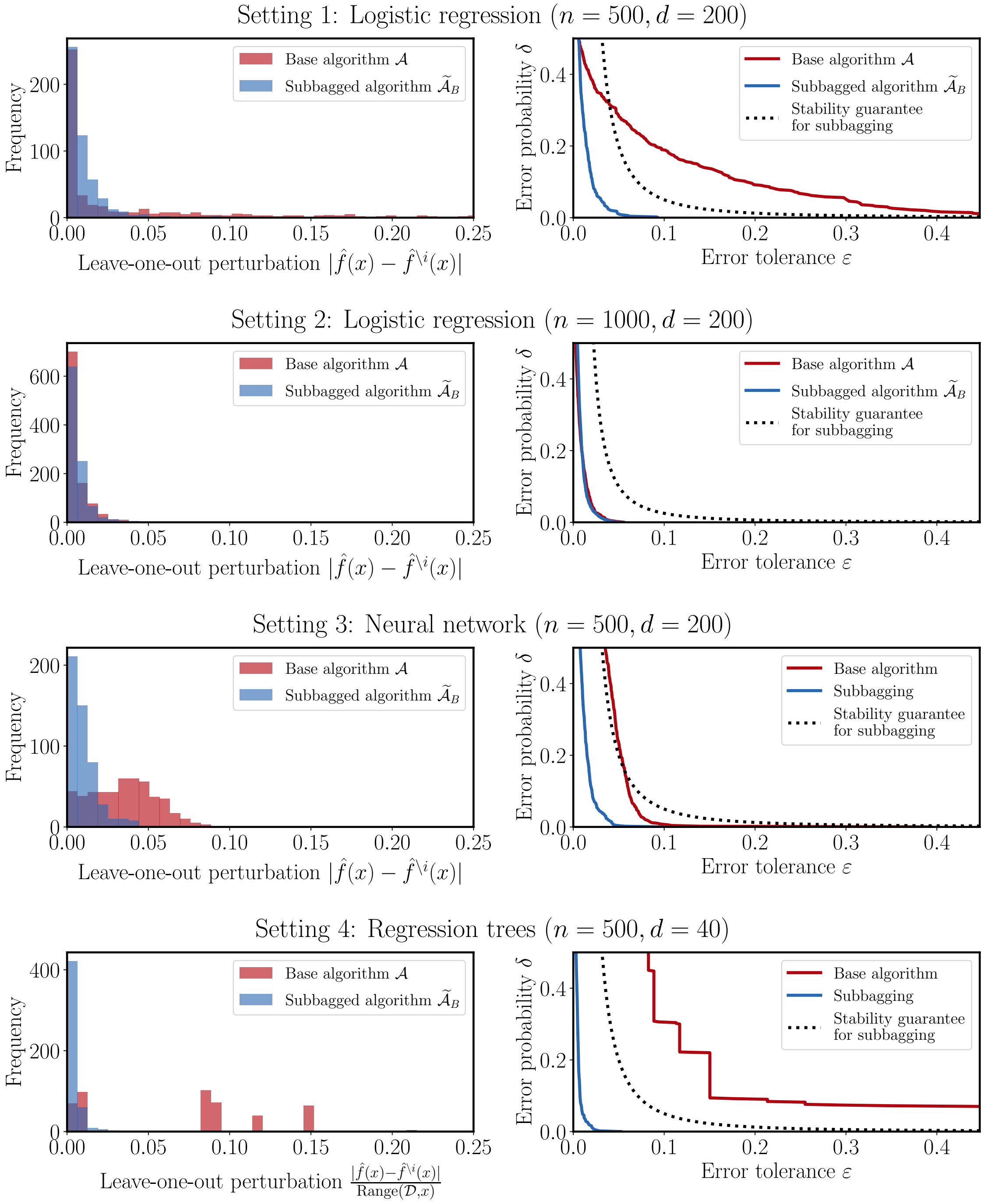

Figure 3: Simulation results comparing the stability of subbagging $\widetilde{\mathcal{A}}_B$ to that of the corresponding base algorithm $\mathcal{A}$. Left: Histogram of leave-one-out perturbations. Right: for each ε, the smallest δ such that the algorithm is (ε, δ)-stable in the sense of Definition 4. Higher curves thus represent greater instability. In all settings, $m = n/2$ and $B = 10000$.

stability by plotting, for each value of ε, the smallest value of δ such that the algorithm is (ε, δ)-stable:

$$\delta = \frac{1}{n} \sum_{i=1}^{n} 1_{|\hat{f}(x) - \hat{f}^{\setminus i}(x)| > \varepsilon}.$$

In each case, the test point $x = X_{n+1}$ is generated from the same distribution as $X_1, \dots, X_n$.

For logistic regression (Settings 1 and 2), we see that the subbagged algorithm is highly stable for both values of n—in particular, the blue curves lie below the black dotted line, showing that subbagged logistic regression satisfies the theoretical guarantee of Theorem 8. By contrast, for $n = 500$ and $d = 200$ (Setting 1), the red curve lies much higher in the plot, showing greater instability; this reveals that the base algorithm, logistic regression (with extremely small regularization), is highly unstable in this regime (see, e.g., Candès and Sur, 2020). For $n = 1000$ and $d = 200$ (Setting 2), on the other hand, we see that the base algorithm is quite stable—indeed, in this setting, each bag is highly unstable (since $m = n/2 = 500$), but the stability of subbagging is still comparable to that of the base algorithm. These first two settings illustrate our theory by showing that the subbagged algorithm satisfies the stability guarantee regardless of whether the base algorithm is stable.

In Setting 3, we repeat the same experiment where the base algorithm is a neural network. The neural network base algorithm slightly violates the stability guarantee, and in this case, subbagging improves the stability.

In Setting 4, we simulate from a more complex data generating process. We again see that the subbagged algorithm is highly stable—in particular, the blue curve lies below the black dotted line, showing that subbagged regression trees satisfy the theoretical guarantee of Theorem 8. By contrast, the red curve lies much higher in the plot, showing greater instability; this reveals that the base algorithm, a regression tree with a maximum depth of 50, is highly unstable.

7. Discussion and Related Work

In this section, we first discuss some important practical implications of algorithmic stability. Next, we compare our main question to a prior work attempting to certify stability using hypothesis testing (Kim and Barber, 2023). Finally, we situate our work in the broader literature on the stability of bagging, and give some concluding remarks on the implications of this work.

7.1 The Importance of Stability

Stability guarantees are central in a variety of contexts, despite the fact that many widely-used practical algorithms are not stable (Xu et al., 2011). For instance, Bousquet and Elisseeff (2002) establish generalization bounds for stable learning algorithms, and Mukherjee et al. (2006) show that stability is necessary and sufficient for empirical risk minimization to be consistent; related works include (Poggio et al., 2004; Kutin and Niyogi, 2002; Freund et al., 2004). Shalev-Shwartz et al. (2010) identify stability as a necessary and sufficient condition for learnability. Stability is further relevant to differential privacy guarantees; assuming worst-case stability (often called "sensitivity" in the privacy literature) is a standard starting point for constructing differentially private algorithms (Dwork, 2008). In the field of conformal prediction, distribution-free coverage guarantees rely upon the stability of the underlying estimators (e.g., Steinberger and Leeb, 2016, 2023; Ndiaye, 2022; Barber et al., 2021). We now discuss applications of algorithmic stability to generalization and conformal inference in greater detail.

7.1.1 Stability and Generalization

In a landmark work, Bousquet and Elisseeff (2002) greatly expand our understanding of the connection between stability and generalization. In their telling, what distinguishes algorithmic stability from the pervasive uniform convergence theory is the following: whereas the latter aims to control the complexity of the space of learning rules an algorithm $\mathcal{A}$ searches over, the former emphasizes how the algorithm explores that space. Algorithmic stability notably first emerged as an invaluable tool to obtain generalization bounds for k-nearest neighbors (Rogers and Wagner, 1978), for which the underlying function class has unbounded complexity. For algorithms like bagging and nearest neighbors, where the strongest (nontrivial) guarantees hold for out-of-sample stability, the empirical risk is not necessarily reflective of test error and instead generalization holds with respect to the leave-one-out error—that is, the average leave-one-out error is a provably accurate estimate of the expected prediction error,

$$\frac{1}{n}\sum_{i=1}^{n} \ell(\hat{f}^{\setminus i}(X_i), Y_i) \approx \mathbb{E}[\ell(\hat{f}(X), Y)],$$

where the expected value is taken with respect to a new draw of (X, Y) while treating $\hat{f}$ as fixed. For an example of how random hypothesis stability (covered in Corollary 16) leads to polynomial bounds on the generalization error, see Elisseeff et al. (2005, Theorem 9).

7.1.2 Predictive Uncertainty Quantification

Algorithmic stability also plays an important role in the problem of predictive uncertainty quantification. Suppose $(X_i, Y_i)_{i=1}^{n+1}$ are iid draws from an unknown distribution P, and Y_{n+1} is unobserved. We wish to construct a prediction interval $\hat{C}_{n,\alpha} = \hat{C}_{n,\alpha}(X_{n+1})$ (based on the training data $\mathcal{D} = (X_i, Y_i)_{i=1}^{n}$, test covariate X_{n+1} and learning algorithm $\mathcal{A}$) that has guaranteed predictive coverage, that is,

$$\mathbb{P}\left\{Y_{n+1} \in \hat{C}_{n,\alpha}(X_{n+1})\right\} \geq 1 - \alpha, \tag{18}$$

without any restrictions on $\mathcal{A}$ or P. If we wish to use an interval of the form $\hat{C}_{n,\alpha}(x) = [\hat{f}(x) - \hat{c}, \hat{f}(x) + \hat{c}]$, centered at the learning algorithm's prediction $\hat{f}(x)$, a natural approach to calibrating the radius $\hat{c}$ is to use the leave-one-out errors $R_i := |\hat{f}^{\setminus i}(X_i) - Y_i|$ as representative of the test error $|\hat{f}(X_{n+1}) - Y_{n+1}|$. This leads naturally to the classical leave-one-out technique known as the jackknife:

$$\hat{C}^{\mathrm{Jack}}_{n,\alpha}(x) := [\hat{f}(x) - \hat{c}_\alpha, \hat{f}(x) + \hat{c}_\alpha],$$

where $\hat{c}_\alpha := Q_{1-\alpha}\left(\{R_i\}_{i=1}^{n}\right)$ is the $1 - \alpha$ quantile of the leave-one-out errors R_i.

If the base algorithm $\mathcal{A}$ is *unstable*, the leave-one-out errors need not be representative of test error at all. In fact, Barber et al. (2021) construct a pathological example for which that the jackknife has no coverage, that is,

$$\mathbb{P}\left\{Y_{n+1} \in \hat{C}^{\text{Jack}}_{n,\alpha}(X_{n+1})\right\} = 0.$$

Barber et al. (2021) go on to show that if the base algorithm $\mathcal{A}$ is (ε, δ)-stable, then coverage can be restored by inflating the radius to $\hat{c}'_\alpha := \hat{c}_\alpha + \varepsilon$ and running the procedure at level $\alpha' = \alpha - 2\sqrt{\delta}$.

7.2 Is Bagging Needed for Stability?

Various learning algorithms are known to possess stability guarantees, such as k-nearest neighbors (Rogers and Wagner, 1978; Devroye and Wagner, 1979b), some regularized regression methods such as ridge regression (Bousquet and Elisseeff, 2002; Wibisono et al., 2009), and models trained with stochastic gradient descent under smoothness assumptions (Hardt et al., 2016). Restricting to algorithms that are theoretically known to be stable can be quite limiting and can sacrifice accuracy in many settings.

We might instead ask whether it is possible to validate empirically that an algorithm $\mathcal{A}$ is stable with respect to a given data generating distribution. However, Kim and Barber (2023) show that it is essentially impossible to construct powerful hypothesis tests certifying (ε, δ)-stability, without imposing assumptions on the algorithm or on distribution of the data. In their framework, we observe iid random variables $\mathcal{D} = (Z_i)_{i=1}^N$ where $Z_i \overset{\text{iid}}{\sim} P$. We wish to construct a test $\hat{T}$ that returns an answer 1 if we are confident that $\mathcal{A}$ is (ε, δ)-stable, or a 0 otherwise. Suppose we require that $\hat{T}$ obeys the following constraints:

(a) $\hat{T}$ satisfies a universal bound on falsely declaring stability, that is, $\mathbb{P}\left\{\hat{T} = 1\right\} \leq \alpha$ for any $\mathcal{A}$ that is *not* (ε, δ)-stable (with respect to distribution P and sample size n), and

(b) $\hat{T}$ is a *black-box* test (see Kim and Barber, 2023, Definition 2), roughly meaning that $\hat{T}$ is only constructed using zeroth order oracle access to the algorithm $\mathcal{A}$. That is, we may base our accept/reject decision on evaluating the model $\mathcal{A}$ on data $\widetilde{\mathcal{D}}$ that is simulated or resampled from the training data $\mathcal{D}$, and compute predictions at test points x that are generated similarly, an unlimited number of times.

If a test $\hat{T}$ satisfies both properties (a) and (b) with no further assumptions on the distribution P or on the algorithm $\mathcal{A}$, their results imply that the power of $\hat{T}$ is upper bounded by

$$\mathbb{P}\left\{\hat{T} = 1\right\} \leq (1 - \delta)^{-N/n}\alpha,$$

for any $\mathcal{A}$ that *is* (ε, δ)-stable (with respect to distribution P and sample size n). In particular, any universally valid black-box test has low power, unless the available data set size N is far larger than the sample size n for which we want to test stability.

In light of this impossibility result, a natural question is whether it is possible to *convert* any algorithm $\mathcal{A}$ into an (ε, δ)-stable algorithm $\widetilde{\mathcal{A}}$. Our work establishes the possibility of *black-box stabilization*, that is, guaranteeing some quantifiable level of stability with no knowledge of the inner workings of the base algorithm. Our results support the use of bagging in such settings by certifying a certain level of (ε, δ)-stability.

7.3 Prior Work on the Stability of Bagging

Bühlmann and Yu (2002) suggest (sub)bagging is most successful as a smoothing operation, softening hard threshold rules. They measure the instability of a procedure by its asymptotic variance: $\mathcal{A}$ is stable at $x \in \mathcal{X}$ if $\hat{f}(x) \xrightarrow{p} f(x)$ as $n \to \infty$, for some fixed f. For some hard thresholding rules, they show bagging can reduce asymptotic variance. See also Buja and Stuetzle (2000); Friedman and Hall (2007).

Grandvalet (2004, 2006) exposes some limitations of the variance-reduction perspective. In particular, bagging need not reduce variance, and in simple examples its improvement over the base procedure need not relate to the original procedure's variance. Grandvalet illustrates through experiments a robustness property of bagging: highly influential data points are systematically de-emphasized. The role of p in our main result, Theorem 8, underscores Grandvalet's observation that the main stabilizing effect of bagging comes from the removal of high-leverage data points from a certain fraction of bags.

Elisseeff et al. (2005) generalize standard notions of algorithmic stability (Bousquet and Elisseeff, 2002) to randomized algorithms and study (sub)bagging in this context. We can directly compare Corollary 16 to the work of Elisseeff et al. (2005, Proposition 4.4), who also study the random hypothesis stability of subbagging with respect to an L-Lipschitz loss ℓ. Their result shows subbagging satisfies condition (14) at the level

$$\beta = Lp\beta_{\mathcal{A},m}, \tag{19}$$

where $\beta_{\mathcal{A},m}$ denotes the random hypothesis stability of the base algorithm $\mathcal{A}$ on data sets of size m with respect to ℓ_1 loss. A similar result (under stronger assumptions) was obtained earlier by Poggio et al. (2002).

We can interpret this result in two ways. First, if the base algorithm $\mathcal{A}$ is stable, the guarantee (19) suggests that bagging maintains or improves upon stability (similar results have been obtained for boosting; see, e.g., Kutin and Niyogi, 2001). Generally, we expect the stability of the base algorithm to improve with the sample size (i.e., $\beta_{\mathcal{A},m} \geq \beta_{\mathcal{A},n}$ for $m \leq n$), so (19) does not necessarily imply subbagging improves upon the stability of running the base algorithm $\mathcal{A}$ on the full data set. Second, the result of Elisseeff et al. (2005) shows that we can achieve random hypothesis stability $\beta = O(n^{-1/2})$ by taking $p = O(n^{-1/2})$. By contrast, Corollary 16 shows subbagging even half the data ($p = 0.5$) can achieve random hypothesis stability $\beta = O(n^{-1/2})$.

Chen et al. (2022, Theorem 5) consider subbagging when $m = o(\sqrt{n})$ and with iid data. Specializing their result to the case of learning algorithms with bounded outputs, they guarantee worst-case stability at the level $\varepsilon = o(n^{-1/2})$ as long as $B \gg n$. By contrast, our result does not require iid data, and gives a faster rate $\varepsilon = o(n^{-3/4})$ for fixed δ and $m = o(\sqrt{n})$ (as well as results for larger m, e.g., for $m = O(n)$).

7.4 Conclusion

Distribution-free uncertainty quantification yields principled statistical tools which input black-box machine learning models and produce predictions with statistical guarantees, such as distribution-free prediction or calibration. Assumption-free stability is an important addition to this list, with a number of practical implications. Our work establishes assumption-free stability for bagging applied to any base algorithm with bounded outputs. These results suggest several avenues for future investigations, including formalizing lower bounds for distribution-free, black-box stabilization and characterizing the (sub)optimality of bagging.

Acknowledgments

RFB was supported by the National Science Foundation via grants DMS-1654076 and DMS-2023109, and by the Office of Naval Research via grant N00014-20-1-2337. JAS was supported by NSF DMS-2023109. RW was supported by NSF DMS-2023109, AFOSR FA9550-18-1-0166, NSF DMS-AWD00000326 and Simons Foundation MP-TMPS-00005320.

Appendix A. Proofs

This section contains proofs of all theoretical results from the main paper.

A.1 Proof of Theorem 8

We abbreviate predicted values using $\hat{y} := \hat{f}_\infty(x)$ and $\hat{y}^{\setminus i} := \hat{f}_\infty^{\setminus i}(x)$. Define

$$\mathcal{K} := \left\{ i \in [n] : \left| \hat{y} - \hat{y}^{\setminus i} \right| > \varepsilon \right\},$$

the set of data points with large leave-one-out perturbation, and let $K = |\mathcal{K}|$. From Definition 4, we want to show $K \leq n\delta$. Summing all the inequalities defining $\mathcal{K}$ gives

$$K\varepsilon \leq \sum_{i \in \mathcal{K}} \left| \hat{y} - \hat{y}^{\setminus i} \right| =: L_1(\mathcal{K}).$$

We now bound the error $L_1(\mathcal{K})$ on the right-hand side.

Step 1: Simplifying the leave-one-out perturbation. For any bag $r \in \mathrm{seq}_{[n]}$, denote the value of our prediction using data $\mathcal{D}_r$ by

$$\hat{y}^{(r)} := \mathbb{E}_\xi[\mathcal{A}(\mathcal{D}_r;\xi)(x)].$$

The aggregate prediction $\hat{y}$ can be expressed as

$$\hat{y} = \mathbb{E}_{r \sim \mathcal{Q}_n, \xi}\left[\mathcal{A}(\mathcal{D}_r;\xi)(x)\right] = \mathbb{E}_{r \sim \mathcal{Q}_n, \xi}\left[\hat{y}^{(r)}\right],$$

while $\hat{y}^{\setminus i}$ can be expressed as

$$\hat{y}^{\setminus i} = \mathbb{E}_{r \sim \mathcal{Q}_{n-1}, \xi}\left[\mathcal{A}((\mathcal{D}^{\setminus i})_r;\xi)(x)\right] = \mathbb{E}_{r \sim \mathcal{Q}_n}\left[\hat{y}^{(r)} \,\middle|\, i \notin r\right],$$

where the last step holds by symmetry (Assumption 5). Using the definition of conditional expectation, we have

$$\begin{aligned} \hat{y} - \hat{y}^{\setminus i} &= \mathbb{E}_{r \sim \mathcal{Q}_n}\left[\hat{y} - \hat{y}^{(r)} \,\middle|\, i \notin r\right] \\ &= \frac{1}{1-p}\mathbb{E}_{r \sim \mathcal{Q}_n}\left[(\hat{y} - \hat{y}^{(r)})\mathbf{1}\{i \notin r\}\right]. \end{aligned}$$

Step 2: Expressing $L_1(\mathcal{K})$ as an expectation. Define

$$s_i := \operatorname{sign}(\hat{y} - \hat{y}^{\setminus i}) \cdot \mathbf{1}_{i \in \mathcal{K}}.$$

For each $i \in \mathcal{K}$, $|\hat{y} - \hat{y}^{\setminus i}| = s_i(\hat{y} - \hat{y}^{\setminus i})$, so by Step 1,

$$L_1(\mathcal{K}) = \frac{1}{1-p} \mathbb{E}_{r \sim \mathcal{Q}_n} \left[(\hat{y} - \hat{y}^{(r)}) \sum_i s_i \mathbf{1}_{i \notin r} \right]. \tag{20}$$

Step 3: Bounding $L_1(\mathcal{K})$. Since $\hat{y}^{(r)}$ has mean $\hat{y}$, we may rewrite the right-hand side of (20) as

$$L_1(\mathcal{K}) = \frac{1}{1-p} \mathbb{E}_{r \sim \mathcal{Q}_n} \left[(\hat{y} - \hat{y}^{(r)}) \left(\sum_i s_i \mathbf{1}_{i \notin r} - \mathbb{E} \left[\sum_i s_i \mathbf{1}_{i \notin r} \right] \right) \right].$$

Applying Cauchy–Schwarz,

$$L_1(\mathcal{K}) \leq \frac{1}{2(1-p)} \sqrt{\operatorname{Var} \left(\sum_i s_i \mathbf{1}_{i \notin r} \right)}, \tag{21}$$

where we have used $\operatorname{Var}(\hat{y}^{(r)}) \leq \frac{1}{4}$, known as Popoviciu's inequality, which uses $\hat{y}^{(r)} \in [0,1]$. We calculate the other variance term as

$$\begin{aligned}
\operatorname{Var} \left(\sum_i s_i \mathbf{1}_{i \notin r} \right) &= p(1-p) \sum_i s_i^2 - q \sum_{i \neq j} s_i s_j \\
&= (p(1-p) + q) \sum_i s_i^2 - q \Big(\sum_i s_i \Big)^2 \\
&\leq K \left(p(1-p) + q \right),
\end{aligned}$$

since $q \geq 0$ and $\sum_i s_i^2 = \sum_i \mathbf{1}_{i \in \mathcal{K}} = K$. Combining everything,

$$K\varepsilon \leq \frac{1}{2(1-p)} \sqrt{K(p(1-p)+q)}.$$

Choosing (ε, δ) to satisfy (7) implies $K \leq n\delta$.

A.2 Proof of Theorem 9

Let $\hat{f}_B$ denote the result of bagging $\mathcal{A}$ on $\mathcal{D}$ with B bags, and similarly $\hat{f}_B^{\setminus i}$ is the result of bagging $\mathcal{A}$ on $\mathcal{D}^{\setminus i}$ with B bags. We want to show

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{P}_{\boldsymbol{\xi}, \boldsymbol{r}} \left\{ |\hat{f}_B(x) - \hat{f}_B^{\setminus i}(x)| > \varepsilon + \sqrt{\frac{2}{B} \log \frac{4}{\delta'}} \right\} \leq \delta + \delta',$$

where $\boldsymbol{\xi} = (\xi^{(1)}, \ldots, \xi^{(B)})$ and $\boldsymbol{r} = (r^{(1)}, \ldots, r^{(B)})$ capture the randomness in the algorithm $\mathcal{A}$ and bagging, respectively. By the triangle inequality and union bound,

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} & \mathbb{P}_{\boldsymbol{\xi},\boldsymbol{r}}\left\{|\hat{f}_B(x) - \hat{f}_B^{\setminus i}(x)| > \varepsilon + \sqrt{\frac{2}{B}\log\frac{4}{\delta'}}\right\} \\
&\leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{P}_{\boldsymbol{\xi},\boldsymbol{r}}\left\{|\hat{f}_B(x) - \hat{f}_\infty(x)| > \sqrt{\frac{1}{2B}\log\frac{4}{\delta'}}\right\} \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left\{|\hat{f}_\infty(x) - \hat{f}_\infty^{\setminus i}(x)| > \varepsilon\right\} \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\mathbb{P}_{\boldsymbol{\xi},\boldsymbol{r}}\left\{|\hat{f}_\infty^{\setminus i}(x) - \hat{f}_B^{\setminus i}(x)| > \sqrt{\frac{1}{2B}\log\frac{4}{\delta'}}\right\}.
\end{aligned}$$

By Theorem 8 the middle term on the right-hand side is at most δ. By Hoeffding's inequality,

$$\mathbb{P}_{\boldsymbol{\xi},\boldsymbol{r}}\left\{|\hat{f}_B(x) - \hat{f}_\infty(x)| > \sqrt{\frac{1}{2B}\log\frac{4}{\delta'}}\right\} \leq \frac{\delta'}{2}$$

and similarly for $|\hat{f}_B^{\setminus i}(x) - \hat{f}_\infty^{\setminus i}(x)|$.

A.3 Proof of Theorem 10

Let $K = 1 + \lfloor \delta n \rfloor$ and $\eta = \frac{K}{n} > \delta$. For any positive integer m, define

$$\mathcal{A}^\sharp\Big((X_1, Y_1), \ldots, (X_m, Y_m)\Big)(x) := \mathbf{1}\left\{\sum_{i=1}^{m} X_i > \frac{mK}{n}\right\}.$$

Define $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where $x_i = 1$ for $i \leq K$ and $x_i = 0$ for $i > K$. For a bag r consisting of m indices sampled without replacement, the algorithm $\mathcal{A}^\sharp$ therefore returns the prediction $\hat{y}(r) = \mathbf{1}\{\sum_{i\in r} X_i > \frac{mK}{n}\}$. Let $\hat{y}$ denote the average prediction, i.e., the result of subbagging $\mathcal{A}^\sharp$, and let $\hat{y}^{\setminus i}$ denote the average prediction over bags excluding i. It suffices to show that $|\hat{y} - \hat{y}^{\setminus i}| > \varepsilon$ for each i with $x_i = 1$, since we then have

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left\{|\hat{y} - \hat{y}^{\setminus i}| > \varepsilon\right\} \geq \frac{K}{n} = \eta > \delta,$$

verifying that $\mathcal{A}^\sharp$ fails to be (ε, δ)-stable. Let $\hat{y}^i$ denote the average over all bags containing i. Then for any $i \leq K$ and $j > K$ (i.e., $X_i = 1$ and $X_j = 0$), by symmetry we can calculate

$$\begin{aligned}
\hat{y} &= \mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r)] \\
&= \frac{K}{n}\mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r) \mid i_1 \leq K] + \frac{n-K}{n}\mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r) \mid i_1 > K] \\
&= \eta\mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r) \mid i_1 = i] + (1-\eta)\mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r) \mid i_1 = j] \\
&= \eta\mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r) \mid i \in r] + (1-\eta)\mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r) \mid j \in r] \\
&= \eta\hat{y}^i + (1-\eta)\,\hat{y}^j.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}\hat{y} &= \mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r)]\\ &= p\mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r) \mid i\in r] + (1-p)\mathbb{E}_{r\sim\mathcal{Q}_n}[\hat{y}(r) \mid i\notin r]\\ &= p\hat{y}^i + (1-p)\hat{y}^{\setminus i}.\end{aligned}$$

Combining these calculations,

$$\hat{y}-\hat{y}^{\setminus i} = \frac{p}{1-p}(\hat{y}^i-\hat{y}) = \frac{p}{1-p}\,(1-\eta)\,(\hat{y}^i-\hat{y}^j).$$

Similarly, noting that $\mathbb{P}\{j\in r \mid i \in r\} = \frac{m-1}{n-1}$, we have

$$\hat{y}^i = \frac{m-1}{n-1}\hat{y}^{ij} + \frac{n-m}{n-1}\hat{y}^{i\setminus j},$$

where $\hat{y}^{ij}$ averages $\hat{y}(r)$ over bags containing both i and j, and similarly $\hat{y}^{i\setminus j}$ averages over bags containing i and not j. Similarly, $\hat{y}^j = \frac{m-1}{n-1}\hat{y}^{ij} + \frac{n-m}{n-1}\hat{y}^{j\setminus i}$, and therefore, $\hat{y}^i - \hat{y}^j = \frac{n-m}{n-1}(\hat{y}^{i\setminus j} - \hat{y}^{j\setminus i})$. We write $\hat{y}^{i\setminus j}$ and $\hat{y}^{j\setminus i}$ as hypergeometric tail probabilities:

$$\begin{aligned}\hat{y}^{i\setminus j} &= \mathbb{P}_{H\sim\mathrm{HyperGeometric}(n-2,K-1,m-1)}\left\{1+H>\frac{mK}{n}\right\}\\ \hat{y}^{j\setminus i} &= \mathbb{P}_{H\sim\mathrm{HyperGeometric}(n-2,K-1,m-1)}\left\{H>\frac{mK}{n}\right\}.\end{aligned}$$

Combining our findings,

$$\begin{aligned}\hat{y}-\hat{y}^{\setminus i} &= \frac{p\,(1-\eta)}{1-p}\frac{n-m}{n-1}(\hat{y}^{i\setminus j}-\hat{y}^{j\setminus i})\\ &= \frac{m\,(1-\eta)}{n-1}\mathbb{P}_{H\sim\mathrm{HyperGeometric}(n-2,K-1,m-1)}\left\{H=\left\lfloor\frac{mK}{n}\right\rfloor\right\}\end{aligned}$$

Now let $h=\lfloor\frac{mK}{n}\rfloor$. Recalling $\eta=\frac{K}{n}$,

$$\begin{aligned}\hat{y}-\hat{y}^{\setminus i} &= \frac{m\,(1-\eta)}{n-1}\mathbb{P}_{H\sim\mathrm{HyperGeometric}(n-2,K-1,m-1)}\left\{H=h\right\}\\ &= \frac{m\,(1-\eta)}{n-1}\frac{\binom{K-1}{h}\binom{n-K-1}{m-h-1}}{\binom{n-2}{m-1}}\\ &= \frac{m\,(1-\eta)}{n-1}\frac{\binom{K-1}{h}\binom{n-K}{m-h}\frac{m-h}{n-K}}{\binom{n-1}{m}\frac{m}{n-1}}\\ &= \frac{m-h}{n}\frac{\binom{K-1}{h}\binom{n-K}{m-h}}{\binom{n-1}{m}}\\ &= \frac{m-\lfloor\frac{mK}{n}\rfloor}{n}\mathbb{P}_{H\sim\mathrm{HyperGeometric}(n-1,K-1,m)}\left\{H=\left\lfloor\frac{mK}{n}\right\rfloor\right\}\\ &\geq (1-\delta-n^{-1})p\mathbb{P}_{H\sim\mathrm{HyperGeometric}(n-1,K-1,m)}\left\{H=\lfloor p(1+\lfloor n\delta\rfloor)\rfloor\right\}\\ &> \varepsilon,\end{aligned}$$

where the last step holds by assumption on ε. This verifies that (ε,δ)-stability fails to hold, and thus completes the proof.

A.4 Proof of Theorem 12

This result follows from the proof of Theorem 8 if we substitute Equation (21) with

$$L_1(\mathcal{K}) \le \frac{R^*(\mathcal{D},x)}{2(1-p)}\sqrt{\mathrm{Var}\left(\sum_i s_i \mathbf{1}_{i\notin r}\right)}.$$

A.5 Proof of Theorem 13

By the triangle inequality,

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^n \mathbf{1}\left\{|\hat{f}_{\infty,I}(x) - \hat{f}^{\setminus i}_{\infty,I}(x)| > \varepsilon\,\mathrm{length}(I(\mathcal{D}))\right\}\\ &\qquad \le \frac{1}{n}\sum_{i=1}^n \mathbf{1}\left\{\frac{|\hat{f}_{\infty,I}(x) - \hat{f}^{\setminus i}_{\infty,I}(x)|}{\mathcal{R}(\mathcal{D},x)} > \varepsilon, I(\mathcal{D}) = I(\mathcal{D}^{\setminus i})\right\}\\ &\qquad\quad + \frac{1}{n}\sum_{i=1}^n \mathbf{1}\left\{I(\mathcal{D}) \neq I(\mathcal{D}^{\setminus i})\right\}.\end{aligned}$$

It suffices to show that the first term on the right-hand side is at most δ.

Let $I_0 := I(\mathcal{D})$ denote the interval based on the full data set. Define a new algorithm $\mathcal{A}^*$ that clips the output to I_0 regardless of the input data set. That is, for any data set $\mathcal{D}'$ and test point $x \in \mathcal{X}$,

$$\mathcal{A}^*(\mathcal{D}')(x) := \frac{\mathbb{E}_\xi\left[\mathrm{Clip}_{I_0}(\mathcal{A}(\mathcal{D}';\xi)(x))\right] - \inf I_0}{\mathrm{length}(I_0)}.$$

This modified base algorithm has bounded outputs—that is, $\mathcal{A}^*(\mathcal{D}')(x) \in [0,1]$. Let $\hat{f}^*_\infty = \widetilde{\mathcal{A}}^*_\infty(\mathcal{D})$ denote the result of derandomized bagging (Algorithm 2) on the modified base algorithm $\mathcal{A}^*$, and similarly let $\hat{f}^{*\setminus i}_\infty = \widetilde{\mathcal{A}}^*_\infty(\mathcal{D}^{\setminus i})$. For any $i \in [n]$, on the event $I(\mathcal{D}) = I(\mathcal{D}^{\setminus i})$, we have

$$\frac{|\hat{f}_{\infty,I}(x) - \hat{f}^{\setminus i}_{\infty,I}(x)|}{\mathcal{R}(\mathcal{D},x)} = |\hat{f}^*_\infty(x) - \hat{f}^{*\setminus i}_\infty(x)|.$$

Hence, applying Theorem 8 to the modified base algorithm $\mathcal{A}^*$,

$$\frac{1}{n}\sum_{i=1}^n \mathbf{1}\left\{\frac{|\hat{f}_{\infty,I}(x) - \hat{f}^{\setminus i}_{\infty,I}(x)|}{\mathcal{R}(\mathcal{D},x)} > \varepsilon, I(\mathcal{D}) = I(\mathcal{D}^{\setminus i})\right\} \le \frac{1}{n}\sum_{i=1}^n \mathbf{1}\left\{|\hat{f}^*_\infty(x) - \hat{f}^{*\setminus i}_\infty(x)| > \varepsilon\right\} \le \delta,$$

completing the proof.

A.6 Proof of Theorem 14

(i) Fix $\mathcal{A}, \mathcal{D}, x$, and let $\hat{f}^i_\infty = \mathbb{E}_{r\sim\mathcal{Q}_n,\xi}[\hat{f}^{(r)} \mid i \in r]$. Observe that $\hat{f} = p\hat{f}^i_\infty + (1-p)\hat{f}^{\setminus i}_\infty$, so

$$|\hat{f}_\infty(x) - \hat{f}^{\setminus i}_\infty(x)| = p|\hat{f}^i_\infty(x) - \hat{f}^{\setminus i}_\infty(x)| \le p$$

for all i. This proves $(p,0)$-stability, and therefore, (p,δ)-stability holds for all δ.

(ii) Let $\mathcal{A}^\dagger(\mathcal{D})(x) = \mathbf{1}\{\exists(\tilde{x},\tilde{y}) \in \mathcal{D} : \tilde{x} = x\}$—i.e., the algorithm checks whether x belongs to the training bag. Take $\mathcal{D} = (x_i, y_i)_{i=1}^n$ such that every x_i is unique. Then, for each i, $\hat{f}_\infty^{\setminus i}(x_i) = 0$, whereas $\hat{f}_\infty(x_i) = p$.

A.7 Proof of Corollary 15

Integrating Hoeffding's inequality,

$$\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{r}}|\hat{f}_B(x) - \hat{f}_\infty(x)| \leq \int_0^1 2\exp\left(-2Bt^2\right) \mathrm{d}t \leq \sqrt{\frac{\pi}{2B}},$$

and similarly for $|\hat{f}_B^{\setminus i}(x) - \hat{f}_\infty^{\setminus i}(x)|$.

By Theorem 18 (given below),

$$\frac{1}{n}\sum_{i=1}^n \left|\hat{f}_\infty(x) - \hat{f}_\infty^{\setminus i}(x)\right| \leq \sqrt{\frac{1}{4n}\left(\frac{p}{1-p} + \frac{q}{(1-p)^2}\right)}.$$

Combining these bounds via the triangle inequality completes the proof.

A.8 Proof of Corollary 16

Since the loss is L-Lipschitz,

$$\begin{aligned}
&\mathbb{E}_{(Z_i)_{i=1}^{n+1} \overset{\text{iid}}{\sim} P, \boldsymbol{\xi},\boldsymbol{r}} \left|\ell(\hat{f}_B(X_{n+1}), Y_{n+1}) - \ell(\hat{f}_B^{\setminus i}(X_{n+1}), Y_{n+1})\right| \\
&\qquad \leq L\,\mathbb{E}_{(Z_i)_{i=1}^{n+1} \overset{\text{iid}}{\sim} P, \boldsymbol{\xi},\boldsymbol{r}} \left|\hat{f}_B(X_{n+1}) - \hat{f}_B^{\setminus i}(X_{n+1})\right| \\
&\qquad = L\,\mathbb{E}_{(Z_i)_{i=1}^{n+1} \overset{\text{iid}}{\sim} P} \left[\frac{1}{n}\sum_{j=1}^n \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{r}} \left|\hat{f}_B(X_{n+1}) - \hat{f}_B^{\setminus j}(X_{n+1})\right|\right]
\end{aligned}$$

The result follows upon applying Corollary 15.

Appendix B. Stability Guarantee in ℓ_k

In the main text, we establish guarantees for both worst-case and average-case stability. These two notions can be viewed as the ℓ_∞ and ℓ_1 norms (respectively) of the sequence of leave-one-out perturbations $(|\hat{f}(x) - \hat{f}^{\setminus i}(x)|)_{i=1}^n$. Our next result interpolates between these two settings by providing a guarantee for the ℓ_k norm for any $k > 0$. We state the result in the derandomized case for simplicity.

Theorem 18 *In the setting of Theorem 8, define*

$$C(\mathcal{Q}_n) := \min\left\{\sqrt{\frac{1}{4n}\left(\frac{p}{1-p} + \frac{q}{(1-p)^2}\right)}, p\right\}.$$

Suppose $C(\mathcal{Q}_n) \leq p$. Then, for any $k > 0$, derandomized bagging $\widetilde{\mathcal{A}}_\infty$ satisfies

$$\left(\frac{1}{n}\sum_{i=1}^n \left|\hat{f}_\infty(x) - \hat{f}_\infty^{\setminus i}(x)\right|^k\right)^{1/k} \leq C(\mathcal{Q}_n)^{2/\max\{k,2\}} p^{1-2/\max\{k,2\}}$$

This result interpolates between some of our main stability guarantees. For instance, as $k \to \infty$, Theorem 18 yields

$$\max_{i=1,\dots,n} \left| \hat{f}_\infty(x) - \hat{f}^{\setminus i}_\infty(x) \right| \le p,$$

recovering part *(i)* of Theorem 14. Corollary 15 covers the special case $k = 1$

$$\frac{1}{n} \sum_{i=1}^n \left| \hat{f}_\infty(x) - \hat{f}^{\setminus i}_\infty(x) \right| \le C(\mathcal{Q}_n),$$

in the derandomized setting ($B \to \infty$). Finally, setting $k = 2$, by Markov's inequality,

$$\frac{1}{n} \sum_{i=1}^n \mathbf{1} \left\{ \left| \hat{f}_\infty(x) - \hat{f}^{\setminus i}_\infty(x) \right| \ge \varepsilon \right\} \le \frac{C^2(\mathcal{Q}_n)}{\varepsilon^2},$$

recovering Theorem 8.

Proof We first prove the result for $k = 2$. We use the same notation as in the proof of Theorem 8, additionally defining $L_i = \left| \hat{y} - \hat{y}^{\setminus i} \right|$ and $s_i = \operatorname{sign}\left(\hat{y} - \hat{y}^{\setminus i} \right)$. Following the same line of reasoning as in the proof of Theorem 8,

$$\begin{aligned}
\|\vec{L}\|_2^2 &= \sum_{i=1}^n s_i L_i \cdot \left(\hat{y} - \hat{y}^{\setminus i} \right) \\
&= \sum_{i=1}^n s_i L_i \cdot \mathbb{E}_r \left[\hat{y} - \hat{y}^{(r)} \middle| i \notin r \right] \\
&= \mathbb{E}_r \left[\frac{1}{1-p} \sum_{i=1}^n s_i L_i \cdot \left(\hat{y} - \hat{y}^{(r)} \right) \mathbf{1}_{i \notin r} \right] \\
&= \mathbb{E}_r \left[\left(\hat{y} - \hat{y}^{(r)} \right) \frac{1}{1-p} \sum_{i=1}^n s_i L_i \cdot \left(\mathbf{1}_{i \notin r} - p \right) \right] \\
&\le \sqrt{ \operatorname{Var}_r \left[\hat{y}^{(r)} \right] \cdot \operatorname{Var}_r \left[\frac{1}{1-p} \sum_{i=1}^n s_i L_i \cdot \left(\mathbf{1}_{i \notin r} - p \right) \right] } \\
&\le \frac{1}{2(1-p)} \sqrt{ \operatorname{Var}_r \left[\sum_{i=1}^n s_i L_i \mathbf{1}_{i \notin r} \right] }.
\end{aligned}$$

Expanding the variance term,

$$\begin{aligned}
\operatorname{Var}\left(\sum_i s_i L_i \mathbf{1}_{i \notin r} \right) &= p(1-p) \sum_i s_i^2 L_i^2 - q \sum_{i \ne j} s_i L_i s_j L_j \\
&= \left(p(1-p) + q \right) \sum_i s_i^2 L_i^2 - q \Big(\sum_i s_i L_i \Big)^2 \\
&\le \left(p(1-p) + q \right) \sum_i L_i^2.
\end{aligned}$$

After some rearranging, we have $\sqrt{\frac{1}{n}\sum_{i=1}^n L_i^2} \leq \sqrt{\frac{1}{4n}\left(\frac{p}{1-p} + \frac{q}{(1-p)^2}\right)}$. By part *(i)* of Theorem 14, we have $\sqrt{\frac{1}{n}\sum_{i=1}^n L_i^2} \leq C(\mathcal{Q}_n)$. Since $\left(\frac{1}{n}\sum_{i=1}^n L_i^k\right)^{1/k}$ is monotone in k, this also implies the result for $k \leq 2$. For $k > 2$, we again use $\sqrt{\frac{1}{n}\sum_{i=1}^n L_i^2} \leq C(\mathcal{Q}_n)$ and $\max_i L_i \leq p$:

$$\left(\frac{1}{n}\sum_{i=1}^n L_i^k\right)^{1/k} \leq \left(\frac{1}{n}\sum_{i=1}^n L_i^2 p^{k-2}\right)^{1/k} \leq C(\mathcal{Q}_n)^{2/k} p^{1-2/k},$$

completing the proof. ■

Appendix C. Unbounded Outputs with Finite B

In this section, we present analogous results to Theorems 12 and 13 for the finite B case.

Theorem 19 *Let $\hat{\mathcal{Y}} = \mathbb{R}$. Fix a distribution $\mathcal{Q}_n$ on $\mathrm{seq}_{[n]}$ satisfying Assumptions 5 and 7, and let $\mathcal{Q}_{n-1}$ be defined as in* (6)*. Let (ε, δ) satisfy Equation* (7) *and fix $\delta' > 0$. For any algorithm $\mathcal{A}$, generic bagging $\widetilde{\mathcal{A}}_B$ is $\left(\varepsilon + \sqrt{\frac{2}{B}\log\left(\frac{4}{\delta'}\right)}, \delta + \delta', \bar{\mathcal{R}}\right)$-stable, where*

$$\bar{\mathcal{R}}(\mathcal{D}, x) := \sup_{r \in \mathrm{seq}_{[n]}, \xi \in [0,1]} \mathcal{A}(\mathcal{D}_r; \xi)(x) - \inf_{r \in \mathrm{seq}_{[n]}, \xi \in [0,1]} \mathcal{A}(\mathcal{D}_r; \xi)(x). \tag{22}$$

Theorem 19 is proved the same way as Theorem 9, where we apply Theorem 12 instead of Theorem 8. Next, we present our result for adaptively clipped bagging in the finite-B regime.

Theorem 20 *Let $\hat{\mathcal{Y}} = \mathbb{R}$. Fix a distribution $\mathcal{Q}_n$ on $\mathrm{seq}_{[n]}$ satisfying Assumptions 5 and 7, and let $\mathcal{Q}_{n-1}$ be defined as in* (6)*. Suppose the mapping to intervals I satisfies*

$$\frac{1}{n}\sum_{i=1}^n \mathbf{1}\{I(\mathcal{D}) \neq I(\mathcal{D}^{\setminus i})\} \leq \delta_I. \tag{23}$$

Let $\mathcal{R}(\mathcal{D}, x) = \mathrm{length}(I(\mathcal{D}))$, let (ε, δ) satisfy Equation (7) *and fix $\delta' > 0$. For any algorithm $\mathcal{A}$, adaptively clipped bagging $\widetilde{\mathcal{A}}_{B,I}$ is $\left(\varepsilon + \sqrt{\frac{2}{B}\log\left(\frac{4}{\delta'}\right)}, \delta_I + \delta + \delta', R\right)$-stable.*

Proof As in the proof of Theorem 13,

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n \mathbb{P}\left\{|\hat{f}_{B,I}(x) - \hat{f}_{B,I}^{\setminus i}(x)| > \varepsilon\,\mathcal{R}(\mathcal{D}, x)\right\} \\
&\qquad \leq \delta_I + \frac{1}{n}\sum_{i=1}^n \mathbb{P}\left\{\frac{|\hat{f}_{B,I}(x) - \hat{f}_{B,I}^{\setminus i}(x)|}{\mathcal{R}(\mathcal{D}, x)} > \varepsilon, I(\mathcal{D}) = I(\mathcal{D}^{\setminus i})\right\}.
\end{aligned}$$

From this point on, following the same arguments as in the proof of Theorem 13, completing the proof via an application of Theorem 9. ■

References

Sameer Agarwal, Henry Milner, Ariel Kleiner, Ameet Talwalkar, Michael Jordan, Samuel Madden, Barzan Mozafari, and Ion Stoica. Knowing when you're wrong: building fast and reliable approximate query processing systems. In *International Conference on Management of Data (SIGMOD)*, pages 481–492, 2014.

Savina Andonova, Andre Elisseeff, Theodoros Evgeniou, and Massimiliano Pontil. A simple algorithm for learning stable machines. In *European Conference on Artificial Intelligence (ECAI)*, pages 513–517, 2002.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Predictive inference with the jackknife+. *Ann. Statist.*, 49(1):486–507, 2021. ISSN 0090-5364,2168-8966. doi: 10.1214/20-AOS1965. URL https://doi.org/10.1214/20-AOS1965.

Sumanta Basu, Karl Kumbier, James B. Brown, and Bin Yu. Iterative random forests to discover predictive and stable high-order interactions. *Proc. Natl. Acad. Sci. USA*, 115 (8):1943–1948, 2018. ISSN 0027-8424,1091-6490. doi: 10.1073/pnas.1711236115. URL https://doi.org/10.1073/pnas.1711236115.

Shai Ben-David, Dávid Pál, and Hans Ulrich Simon. Stability of k-means clustering. In *International Conference on Computational Learning Theory (COLT)*, pages 20–34, 2007.

Olivier Bousquet and André Elisseeff. Stability and generalization. *J. Mach. Learn. Res.*, 2 (3):499–526, 2002. ISSN 1532-4435,1533-7928. doi: 10.1162/153244302760200704. URL https://doi.org/10.1162/153244302760200704.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996a.

Leo Breiman. Heuristics of instability and stabilization in model selection. *Ann. Statist.*, 24 (6):2350–2383, 1996b. ISSN 0090-5364,2168-8966. doi: 10.1214/aos/1032181158. URL https://doi.org/10.1214/aos/1032181158.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Peter Bühlmann and Bin Yu. Analyzing bagging. *Ann. Statist.*, 30(4):927–961, 2002. ISSN 0090-5364,2168-8966. doi: 10.1214/aos/1031689014. URL https://doi.org/10.1214/aos/1031689014.

Andreas Buja and Werner Stuetzle. Smoothing effects of bagging. *Preprint. AT&T Labs-Research*, 2000.

Emmanuel J. Candès and Pragya Sur. The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. *Ann. Statist.*, 48 (1):27–42, 2020. ISSN 0090-5364,2168-8966. doi: 10.1214/18-AOS1789. URL https://doi.org/10.1214/18-AOS1789.

Qizhao Chen, Vasilis Syrgkanis, and Morgane Austern. Debiased machine learning without sample-splitting for stable estimators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Luc P. Devroye and T. J. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Trans. Inform. Theory*, 25(5):601–604, 1979a. ISSN 0018-9448,1557-9654. doi: 10.1109/TIT.1979.1056087. URL https://doi.org/10.1109/TIT.1979.1056087.

Luc P. Devroye and Terry J. Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Trans. Inform. Theory*, 25(2):202–207, 1979b. ISSN 0018-9448,1557-9654. doi: 10.1109/TIT.1979.1056032. URL https://doi.org/10.1109/TIT.1979.1056032.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

Cynthia Dwork. Differential privacy: a survey of results. In *Theory and applications of models of computation*, volume 4978 of *Lecture Notes in Comput. Sci.*, pages 1–19. Springer, Berlin, 2008. ISBN 978-3-540-79227-7; 3-540-79227-9. doi: 10.1007/978-3-540-79228-4_1. URL https://doi.org/10.1007/978-3-540-79228-4_1.

Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. In *International Conference on Computational Learning Theory (COLT)*, pages 1693–1702, 2018.

B. Efron. Bootstrap methods: another look at the jackknife. *Ann. Statist.*, 7(1):1–26, 1979. ISSN 0090-5364,2168-8966. URL http://links.jstor.org/sici?sici=0090-5364(197901)7:1<1:BMALAT>2.0.CO;2-6&origin=MSN.

Andre Elisseeff, Theodoros Evgeniou, and Massimiliano Pontil. Stability of randomized learning algorithms. *J. Mach. Learn. Res.*, 6:55–79, 2005. ISSN 1532-4435,1533-7928.

Yoav Freund, Yishay Mansour, and Robert E. Schapire. Generalization bounds for averaged classifiers. *Ann. Statist.*, 32(4):1698–1722, 2004. ISSN 0090-5364,2168-8966. doi: 10.1214/009053604000000058. URL https://doi.org/10.1214/009053604000000058.

Jerome H. Friedman and Peter Hall. On bagging and nonlinear estimation. *J. Statist. Plann. Inference*, 137(3):669–683, 2007. ISSN 0378-3758,1873-1171. doi: 10.1016/j.jspi.2006.06.002. URL https://doi.org/10.1016/j.jspi.2006.06.002.

Yves Grandvalet. Bagging equalizes influence. *Machine Learning*, 55(3):251–270, 2004.

Yves Grandvalet. Stability of bagged decision trees. In *Scientific Meeting of the Italian Statistical Society (SIS)*, pages 221–230, 2006.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, pages 1225–1234. PMLR, 2016.

Edward F Harrington. Online ranking/collaborative filtering using the perceptron algorithm. In *International Conference on Machine Learning (ICML)*, pages 250–257, 2003.

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

Byol Kim and Rina Foygel Barber. Black-box tests for algorithmic stability. *Information and Inference: A Journal of the IMA*, 12(4):2690–2719, 10 2023. ISSN 2049-8772. doi: 10.1093/imaiai/iaad039. URL `https://doi.org/10.1093/imaiai/iaad039`.

Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael I. Jordan. A scalable bootstrap for massive data. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 76(4):795–816, 2014. ISSN 1369-7412,1467-9868. doi: 10.1111/rssb.12050. URL `https://doi.org/10.1111/rssb.12050`.

Samuel Kutin and Partha Niyogi. The interaction of stability and weakness in AdaBoost. *University of Chicago Department of Computer Science*, 2001.

Samuel Kutin and Partha Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, page 275?282, 2002.

Kasper Green Larsen. Bagging is an optimal PAC learner. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 450–468. PMLR, 2023.

Daniel LeJeune, Hamid Javadi, and Richard Baraniuk. The implicit regularization of ordinary least squares ensembles. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3525–3535. PMLR, 2020.

Daniel LeJeune, Pratik Patil, Hamid Javadi, Richard G. Baraniuk, and Ryan J. Tibshirani. Asymptotics of the sketched pseudoinverse. *SIAM Journal on Mathematics of Data Science*, 6(1):199–225, 2024. doi: 10.1137/22M1530264. URL `https://doi.org/10.1137/22M1530264`.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(4):417–473, 2010. ISSN 1369-7412,1467-9868. doi: 10.1111/j.1467-9868.2010.00740.x. URL `https://doi.org/10.1111/j.1467-9868.2010.00740.x`.

Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Adv. Comput. Math.*, 25(1-3):161–193, 2006. ISSN 1019-7168,1572-9044. doi: 10.1007/s10444-004-7634-z. URL `https://doi.org/10.1007/s10444-004-7634-z`.

Eugene Ndiaye. Stable conformal prediction sets. In *International Conference on Machine Learning (ICML)*, pages 16462–16479. PMLR, 2022.

Nikunj C Oza and Stuart J Russell. Online bagging and boosting. In *International Workshop on Artificial Intelligence and Statistics*, pages 229–236. PMLR, 2001.

Pratik Patil, Jin-Hong Du, and Arun Kumar Kuchibhotla. Bagging in overparameterized learning: Risk characterization and risk monotonization. *J. Mach. Learn. Res.*, 319:1–113, 2023.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Tomaso Poggio, Ryan Rifkin, Sayan Mukherjee, and Alex Rakhlin. Bagging regularizes. Technical report, MIT, 2002.

Tomaso Poggio, Ryan Rifkin, Sayan Mukherjee, and Partha Niyogi. General conditions for predictivity in learning theory. *Nature*, 428(6981):419–422, 2004.

Zhimei Ren, Yuting Wei, and Emmanuel Candès. Derandomizing knockoffs. *J. Amer. Statist. Assoc.*, 118(542):948–958, 2023. ISSN 0162-1459,1537-274X. doi: 10.1080/01621459.2021.1962720. URL https://doi.org/10.1080/01621459.2021.1962720.

W. H. Rogers and T. J. Wagner. A finite sample distribution-free performance bound for local discrimination rules. *Ann. Statist.*, 6(3):506–514, 1978. ISSN 0090-5364,2168-8966. URL http://links.jstor.org/sici?sici=0090-5364(197805)6:3<506:AFSDPB>2.0.CO;2-M&origin=MSN.

Rajen D. Shah and Richard J. Samworth. Variable selection with error control: another look at stability selection. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 75(1):55–80, 2013. ISSN 1369-7412,1467-9868. doi: 10.1111/j.1467-9868.2011.01034.x. URL https://doi.org/10.1111/j.1467-9868.2011.01034.x.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *J. Mach. Learn. Res.*, 11:2635–2670, 2010. ISSN 1532-4435,1533-7928.

Lukas Steinberger and Hannes Leeb. Leave-one-out prediction intervals in linear regression models with many variables. *arXiv preprint arXiv:1602.05801*, 2016.

Lukas Steinberger and Hannes Leeb. Conditional predictive inference for stable algorithms. *Ann. Statist.*, 51(1):290–311, 2023. ISSN 0090-5364,2168-8966. doi: 10.1214/22-aos2250. URL https://doi.org/10.1214/22-aos2250.

Giorgio Valentini and Francesco Masulli. Ensembles of learning machines. In *Italian workshop on neural nets (WIRN)*, pages 3–20, 2002.

Andre Wibisono, Lorenzo Rosasco, and Tomaso Poggio. Sufficient conditions for uniform stability of regularization algorithms. *Computer Science and Artificial Intelligence Laboratory Technical Report, MIT-CSAIL-TR-2009-060*, 2009.

Huan Xu, Constantine Caramanis, and Shie Mannor. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):187–193, 2011.

Bin Yu. Stability. *Bernoulli*, 19(4):1484–1500, 2013. ISSN 1350-7265,1573-9759. doi: 10.3150/13-BEJSP14. URL https://doi.org/10.3150/13-BEJSP14.