Bagging, Subagging and Bragging for Improving some Prediction Algorithms

Peter Bühlmann
Seminar für Statistik, ETH Zürich, CH-8092 Zürich, Switzerland

Bagging (**b**ootstrap **agg**regat**ing**), proposed by Breiman [1], is a method to improve the predictive power of some special estimators or algorithms such as regression or classification trees. First, we review a recently developed theory explaining why bagging decision trees, or also the subagging (**sub**sample **agg**regat**ing**) variant, yields smooth decisions, reducing the variance and mean squared error. We then propose bragging (**b**ootstrap **r**obust **agg**regat**ing**) as a new version of bagging which, in contrast to bagging, is empirically demonstrated to improve also the MARS algorithm which itself already yields continuous function estimates. Finally, bagging is demonstrated as a "module" in conjunction with boosting for an example about tumor classification using microarray gene expressions.

1. Introduction

Bagging [1], a sobriquet for **b**ootstrap **agg**regat**ing**, is a method for improving unstable estimation or classification schemes. It is very useful for large, high dimensional data set problems where finding a good model or classifier is difficult.

Breiman [1] motivated bagging as a variance reduction technique for a given basis algorithm (i.e. an estimator) such as decision trees or a method that does variable selection and fitting in a linear model. It has attracted much attention and is quite frequently applied, although theoretical insights have been lacking until very recently [4–6]. We present here parts of a theory from Bühlmann and Yu [4], indicating that bagging is a smoothing operation which turns out to be advantageous when aiming to improve the predictive performance of regression or classification trees. In case of regression trees, this theory confirms Breiman's intuition that bagging is a variance reduction technique, reducing also the mean squared error (MSE). The same also holds for subagging (**sub**sample **agg**regat**ing**) which is a computationally cheaper version than bagging. However, for other "complex" basis algorithms, the variance and MSE reduction effect of bagging is not necessarily true; this has also been shown by Buja and Stuetzle [5] in the simpler case where the estimator is a U-statistics.

Moreover, we propose bragging (**b**ootstrap **r**obust **agg**regat**ing**) as a simple, yet new modification which improves an estimation procedure not necessarily because of its smoothing effects but also due to averaging over unstable selection of variables or terms in complex models or algorithms; empirical examples are given for bragging MARS, where bagging is often not a useful device, whereas bragging turns out to be effective.

Bagging can also be useful as a "module" in other algorithms: BagBoosting [3] is a boosting algorithm [8] with a bagged learner, often a bagged regression tree. From our theory it will become evident that BagBoosting using bagged regression trees, which have smaller MSEs than trees, is better than boosting with regression trees. We demonstrate improvements of BagBoosting over boosting for a problem about tumor classification using microarray gene expression predictors.

2. Bagging and Subagging

Consider the regression or classification setting. The data is given as i.i.d. pairs (X_i, Y_i) $(i = 1, \ldots, n)$, where $X_i \in \mathbb{R}^d$ denotes the d-dimensional predictor variable and $Y_i \in \mathbb{R}$ (regression) or $Y_i \in \{0, 1, \ldots, J-1\}$ (classification with J classes). The goal is function estimation and the target function is usually $\mathbb{E}[Y|X = x]$ for regression or the multivariate function $\mathbb{P}[Y = j|X = x]$ $(j = 1, \ldots, J-1)$ for classification. We denote in the sequel a function estimator, which is the outcome from a given basis algorithm, by

$$\hat{\theta}_n(\cdot) = h_n((X_1, Y_1), \ldots, (X_n, Y_n))(\cdot) : \ \mathbb{R}^d \to \mathbb{R},$$

defined by the function $h_n(\cdot)$. Examples of such estimators include linear regression with variable selection, regression trees such as CART [2] or MARS [9].

Definition 1 *(Bagging). Theoretically, bagging is defined as follows.*

(I) Construct a bootstrap sample $(X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)$ *by random drawing* n *times with replacement from the data* $(X_1, Y_1), \ldots, (X_n, Y_n)$.

(II) Compute the bootstrapped estimator $\hat{\theta}_n^*(\cdot)$ *by the plug-in principle:* $\hat{\theta}_n^*(\cdot) = h_n((X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*))(\cdot)$.

(III) The bagged estimator is $\hat{\theta}_{n;Bag}(\cdot) = \mathbb{E}^*[\hat{\theta}_n^*(\cdot)]$.

In practice, the bootstrap expectation in (III) is implemented by Monte Carlo: for every bootstrap simulation $b \in \{1, \ldots, B\}$ from (I), we compute $\hat{\theta}_n^{*b}(\cdot)$ $(b = 1, \ldots, B)$ as in (II) to approximate

$$\hat{\theta}_{n;Bag}(\cdot) \approx B^{-1} \sum_{b=1}^{B} \hat{\theta}_n^{*b}(\cdot). \tag{1}$$

The number B is often chosen as 50 or 100, depending on sample size and on the computational cost to evaluate the estimator $\hat{\theta}_n(\cdot)$. The theoretical quantity in (III) corresponds to $B = \infty$: the finite number B in practice governs the accuracy of the Monte Carlo approximation but otherwise, it shouldn't be viewed as a tuning parameter for bagging.

This is exactly Breiman's [1] definition for bagging regression estimators $\hat{\theta}_n(\cdot)$. For classification, we propose to average the bootstrapped probabilities $\hat{\theta}_{n,j}^{*b}(\cdot) = \hat{\mathbb{P}}^*[Y^{*b} = j|X = \cdot]$ $(j = 0, \ldots, J-1)$ yielding an estimator for $\mathbb{P}[Y = j|X = \cdot]$, whereas Breiman [1] proposed to vote among classifiers for constructing the bagged classifier.

A trivial equality indicates the somewhat unusual approach of using the bootstrap methodology:

$$\hat{\theta}_{n;Bag}(\cdot) = \hat{\theta}_n(\cdot) + (\mathbb{E}^*[\hat{\theta}_n^*(\cdot)] - \hat{\theta}_n(\cdot)) = \hat{\theta}_n(\cdot) + \mathrm{Bias}_n^*(\cdot),$$

where $\mathrm{Bias}_n^*(\cdot)$ is the usual bootstrap bias estimate of $\hat{\theta}_n(\cdot)$. Instead of the usual bias correction with a negative sign, bagging comes along with the wrong sign and *adds* the bootstrap bias estimate. Thus, we would expect that bagging has a higher bias than $\hat{\theta}_n(\cdot)$, which we will argue to be true in some sense. But according to the usual interplay between bias and variance in nonparametric statistics, the hope is to gain more by reducing the variance than increasing the bias so that overall, bagging would pay-off in terms of the MSE. Again, this hope turns out to be true for some basis algorithms (or estimation methods). In fact, Breiman [1] describes heuristically the performance of bagging as follows. The variance of the bagged estimator $\hat{\theta}_{n;Bag}(\cdot)$ is equal or smaller than that for the original estimator $\hat{\theta}_n(\cdot)$; and there can be a drastic variance reduction if the original estimator is "unstable".

Breiman [1] only gives a heuristic definition of instability. Bühlmann and Yu [4] define the following asymptotic notion of instability.

Definition 2 *(Stability of an estimator). An estimator $\hat{\theta}_n(x)$ is called stable at x if $\hat{\theta}_n(x) = \theta(x) + o_P(1)$ $(n \to \infty)$ for some fixed value $\theta(x)$.*

Although this definition resembles very much the one for consistency, it is different in spirit since the value $\theta(x)$ here is only a stable limit and not necessarily the parameter of interest. Instability thus takes place whenever the procedure $\hat{\theta}_n(\cdot)$ is not converging to a fixed value: other realizations from the data generating distribution (even for infinite sample size) would produce a different value of the procedure, with positive probability.

2.1. Unstable estimators with hard decision indicator

Instability often occurs when hard decisions with indicator functions are involved as in regression or classification trees. One of the main underlying ideas why bagging works can be demonstrated with a simple example.

2.1.1. Toy example: a simple, instructive analysis

Consider the estimator

$$\hat{\theta}_n(x) = \mathbf{1}_{[\overline{Y}_n \le x]}, \ x \in \mathbb{R}, \tag{2}$$

where $\overline{Y}_n = n^{-1}\sum_{i=1}^n Y_i$ (no predictor variables X_i are used for this example). The target we have in mind is $\theta(x) = \lim_{n\to\infty}\mathbb{E}[\hat{\theta}_n(x)]$. If we take the view of fixed x, after a proper scaling, a simple yet precise analysis below shows that bagging is a smoothing operation. Due to the central limit theorem we have

$$n^{1/2}(\overline{Y}_n - \mu) \to_D \mathcal{N}(0, \sigma^2) \ (n \to \infty) \tag{3}$$

with $\mu = \mathbb{E}[Y_1]$ and $\sigma^2 = \mathrm{Var}(Y_1)$. Then, for x in a $n^{-1/2}$-neighborhood of μ,

$$x = x_n(c) = \mu + c\sigma n^{-1/2}, \tag{4}$$

we have the distributional approximation

$$\hat{\theta}_n(x_n(c)) \to_D g(Z) = \mathbf{1}_{[Z \le c]} \ (n \to \infty), \ Z \sim \mathcal{N}(0,1). \tag{5}$$

Obviously, for a fixed c, this is a hard decision function of Z. Denoting by $\Phi(\cdot)$ the c.d.f. of a standard normal distribution, it follows that

$$\begin{aligned} &\mathbb{E}[\hat{\theta}_n(x_n(c))] \to \mathbb{P}[Z \le c] = \Phi(c) \ (n \to \infty), \\ &\mathrm{Var}(\hat{\theta}_n(x_n(c))) \to \Phi(c)(1-\Phi(c)) \ (n \to \infty). \end{aligned} \tag{6}$$

Since the variance does not converge to zero, $\hat{\theta}_n(x_n(c))$ is unstable in the sense of Definition 2: the predictor takes the values 0 and 1 with a positive probability, even as n tends to infinity. On the other hand, averaging for the bagged estimator looks as follows:

$$\begin{aligned} \hat{\theta}_{n;Bag}(x_n(c)) &= \mathbb{E}^*[\mathbf{1}_{[\overline{Y}_n^* \le x_n(c)]}] = \mathbb{E}^*[\mathbf{1}_{[n^{1/2}(\overline{Y}_n^* - \overline{Y}_n)/\sigma \le n^{1/2}(x_n(c) - \overline{Y}_n)/\sigma]}] \\ &= \Phi(n^{1/2}(x_n(c) - \overline{Y}_n)) + o_P(1) \\ &\to_D g_{Bag}(Z) = \Phi(c - Z) \ (n \to \infty), \ Z \sim \mathcal{N}(0,1), \end{aligned} \tag{7}$$

where the first approximation (second line) follows because the bootstrap works for the arithmetic mean $\overline{Y}_n$, i.e.,

$$\sup_{x \in \mathbb{R}} |\mathbb{P}^*[n^{1/2}(\overline{Y}_n^* - \overline{Y}_n) \le x] - \mathbb{P}[n^{1/2}(\overline{Y}_n - \mu) \le x]| = o_P(1) \ (n \to \infty), \tag{8}$$

and the second approximation (third line in (7)) holds, because of (3) and the definition of $x_n(c)$ in (4). Comparing with (5), bagging produces a soft decision function of Z: it is a shifted inverse probit, similar to a sigmoid-type function. Figure 1 illustrates the two functions $g(\cdot)$ and $g_{Bag}(\cdot)$. We see that bagging is a smoothing operation. The amount of smoothing is determined "automatically" and turns out to be very reasonable (we are not claiming any optimality here). The effect of smoothing is that bagging reduces

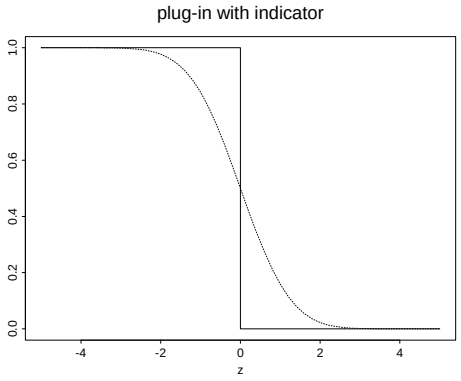

Figure 1. Indicator estimator from (2) at $x = x_n(0)$ as in (4). Function $g(z) = \mathbf{1}_{[z \le 0]}$ (solid line) and $g_{Bag}(z)$ (dotted line) defining the asymptotics of the estimator in (5) and its bagged version in (7).

variance due to a soft- instead of a hard-thresholding operation. An instructive case is with $x = x_n(0) = \mu$, i.e., x is exactly at the most unstable location, where $\mathrm{Var}(\hat{\theta}_n(x))$ is maximal. Formula (7) gives

$$\hat{\theta}_{n;B}(x_n(0)) \to_D \Phi(-Z) = U,\ U \sim \mathrm{Uniform}([0,1]).$$

Thus,

$$\mathbb{E}[\hat{\theta}_{n;B}(x_n(0))] \to \mathbb{E}[U] = 1/2\ (n \to \infty)$$
$$\mathrm{Var}(\hat{\theta}_{n;B}(x_n(0))) \to \mathrm{Var}(U) = 1/12\ (n \to \infty).$$

Comparing with (6), bagging is asymptotically unbiased (for the asymptotic parameter $\lim_{n\to\infty}\mathbb{E}[\hat{\theta}_n(x_n(0))] = \Phi(0) = 1/2$), but the asymptotic variance is reduced by a factor 3!

More generally, we can compute the first two asymptotic moments in the unstable region with $x = x_n(c)$. Denote the convolution of g_1 and g_2 by $g_1 * g_2(\cdot) = \int_{\mathbb{R}} g_1(\cdot - y)g_2(y)dy$, and the standard normal density by $\varphi(\cdot)$.

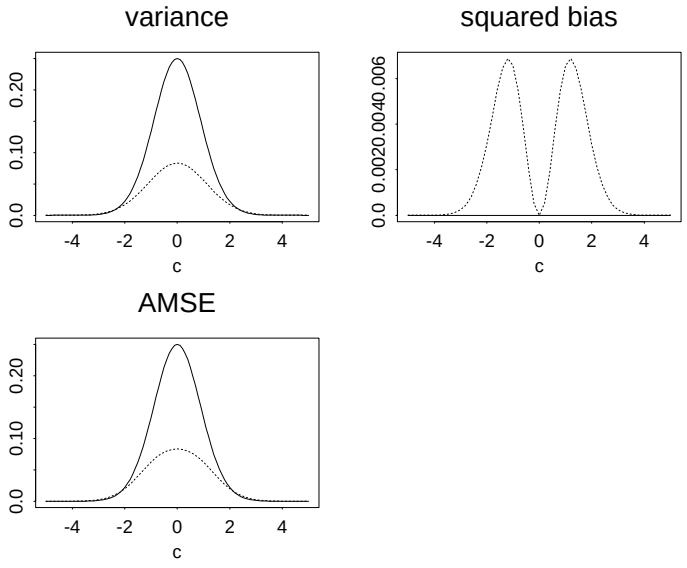

Figure 2. Indicator estimator from (2) at $x = x_n(c)$ as in (4). Asymptotic variance, squared bias and mean squared error (AMSE) (the target is $\lim_{n\to\infty}\mathbb{E}[\hat{\theta}_n(x)]$) for the estimator $\hat{\theta}_n(x_n(c))$ from (2) (solid line) and for the bagged estimator $\hat{\theta}_{n;B}(x_n(c))$ (dotted line) as a function of c.

Corollary 1 *For the estimator in (2) with $x = x_n(c)$ as in formula (4),*

(i) $\lim_{n\to\infty}\mathbb{E}[\hat{\theta}_n(x_n(c))] = \Phi(c)$,
$\lim_{n\to\infty}\mathrm{Var}(\hat{\theta}_n(x_n(c))) = \Phi(c)(1-\Phi(c))$.

(ii) $\lim_{n\to\infty} \mathbb{E}[\hat{\theta}_{n;B}(x_n(c))] = \Phi * \varphi(c)$,
$\lim_{n\to\infty} \mathrm{Var}(\hat{\theta}_{n;B}(x_n(c))) = \Phi^2 * \varphi(c) - (\Phi * \varphi(c))^2$.

Proof: Assertion (i) is restating (6). Assertion (ii) follows by (7) together with the boundedness of the function $g_{Bag}(\cdot)$ therein. □

Numerical evaluations of these first two asymptotic moments and the mean squared error (MSE) are given in Figure 2. We see that in the approximate range where $|c| \leq 2.3$, bagging improves the asymptotic MSE. The biggest gain is at the most unstable point $x = \mu = \mathbb{E}[Y_1]$, corresponding to $c = 0$. The squared bias with bagging has only a negligible effect on the MSE (note the different scales in Figure 2). Note that we always give an a-priori advantage to the original estimator which is asymptotically unbiased for the target as defined.

In [4], this kind of analysis has been given for more general estimators than $\overline{Y}_n$ in (2) and also for estimation in linear models after testing. Hard decision indicator functions are involved there as well and bagging reduces variance due to its smoothing effect. The key to derive this property is always the fact that the bootstrap is asymptotically consistent as in (8).

2.1.2. Regression trees

We address here the effect of bagging in the case of decision trees which are most commonly used in practice in conjunction with bagging. Decision trees consist of piecewise constant fitted functions whose supports (for the piecewise constants) are given by indicator functions similar to (2). Hence we expect bagging to bring a significant variance reduction as in section 2.1.1.

For simplicity of exposition, we consider first a one-dimensional predictor space and a so-called regression stump which is a regression tree with one split and two terminal nodes. The stump estimator (or algorithm) is then defined as the decision tree,

$$\hat{\theta}_n(x) = \hat{\beta}_\ell \mathbf{1}_{[x<\hat{d}_n]} + \hat{\beta}_u \mathbf{1}_{[x\geq\hat{d}_n]} = \hat{\beta}_\ell + (\hat{\beta}_u - \hat{\beta}_\ell)\mathbf{1}_{[\hat{d}_n\leq x]}, \tag{9}$$

where the estimates are obtained by least squares as

$$(\hat{\beta}_\ell, \hat{\beta}_u, \hat{d}_n) = \mathrm{argmin}_{\beta_\ell,\beta_u,d} \sum_{i=1}^{n} (Y_i - \beta_\ell \mathbf{1}_{[X_i<d]} - \beta_u \mathbf{1}_{[X_i\geq d]})^2.$$

These values are estimates for the best projected parameters defined by

$$(\beta_\ell^0, \beta_u^0, d^0) = \mathrm{argmin}_{\beta_\ell,\beta_u,d} \mathbb{E}[(Y - \beta_\ell \mathbf{1}_{[X<d]} - \beta_u \mathbf{1}_{[X\geq d]})^2]. \tag{10}$$

The main mathematical difference of the stump in (9) to the toy estimator in (2) is the behavior of $\hat{d}_n$ in comparison to the behavior of $\overline{Y}_n$ (and not the constants $\hat{\beta}_\ell$ and $\hat{\beta}_u$ involved in the stump). It is shown in [4] that $\hat{d}_n$ has convergence rate $n^{-1/3}$ (in case of a smooth regression function) and a limiting distribution which is non-Gaussian. This also explains that the bootstrap is not consistent, but consistency as in (8) turned out to be crucial in our analysis in section 2.1.1. Summarizing, the asymptotic analysis of bagging a stump is difficult and still unsolved. However, a computationally attractive version of bagging, which has been found as good as bagging, turns out to be more tractable from a theoretical point of view.

2.2. Subagging

Subagging is a sobriquet for **sub**sample **agg**regat**ing** where subsampling is used instead of the bootstrap for the aggregation. An estimator $\hat{\theta}_n(\cdot) = h_n((X_1, Y_1), \ldots, (X_n, Y_n))(\cdot)$ is aggregated as follows:

$$\hat{\theta}_{n;SB(m)}(\cdot) = \binom{n}{m}^{-1} \sum_{(i_1,\ldots,i_m)\in\mathcal{I}} h_m((X_{i_1}, Y_{i_1}), \ldots, (X_{i_m}, Y_{i_m}))(\cdot),$$

where $\mathcal{I}$ is the set of m-tuples whose elements in $\{1, \ldots, n\}$ are all distinct. This aggregation can be approximated by a stochastic computation. The subagging algorithm is as follows.

(I) For $b = 1, \ldots, B$ ($B = 50$ or 100) do:

(i) Generate a random subsample $(X_1^{*b}, Y_1^{*b}), \ldots, (X_m^{*b}, Y_m^{*b})$ by random drawing m times without replacement from the data $(X_1, Y_1), \ldots, (X_n, Y_n)$ (instead of resampling with replacement in bagging).

(ii) Compute the subsampled estimator $\hat{\theta}_m^{*b}(\cdot) = h_m((X_1^{*b}, Y_1^{*b}), \ldots, (X_m^{*b}, Y_m^{*b}))(\cdot)$.

(II) Average the subsampled estimators to approximate $\hat{\theta}_{n;SB(m)}(\cdot) \approx B^{-1} \sum_{b=1}^{B} \hat{\theta}_n^{*b}(\cdot)$.

As indicated in the notation, subagging depends on the subsample size m which is a tuning parameter (in contrast to B).

An interesting case is *half subagging* with $m = [n/2]$. More generally, we could also use $m = [an]$ with $0 < a < 1$ (i.e. m a fraction of n) and we will argue why the usual choice $m = o(n)$ in subsampling for distribution estimation [12] is a bad choice. Half subagging with $m = [n/2]$ has been studied also by Buja and Stuetzle [5]: they showed that in case where $\hat{\theta}_n$ is a U-statistic, half subagging is exactly equivalent to bagging. Moreover, they observe, consistently with our experience, that half subagging yields very similar empirical results to bagging when the estimator $\hat{\theta}_n(\cdot)$ is a decision tree. Thus, if we don't want to optimize over the tuning parameter m, very often a good choice in practice is $m = [n/2]$. Consequently, half subagging typically saves more than half of the computing time because the computational order of an estimator $\hat{\theta}_n$ is usually at least linear in n.

2.2.1. Subagging regression trees

We describe here in a non-technical way the main mathematical result from [4] about subagging regression trees.

The underlying assumptions for our mathematical theory are as follows. The data generating regression model is

$$Y_i = f(X_i) + \varepsilon_i,$$

where $X_1, \ldots, X_n$ and $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. variables, independent from each other, and $\mathbb{E}[\varepsilon_1] = 0$, $\mathbb{E}|\varepsilon_1|^2 < \infty$. The regression function $f(\cdot)$ is assumed to be smooth and the distribution of X_i and ε_i are assumed to have suitably regular densities.

It is then shown in [4] that for $m = [an]$ $(0 < a < 1)$,

$$\limsup_{n\to\infty} \frac{\mathbb{E}[(\hat{\theta}_{n;SB(m)}(x) - \theta(x))^2]}{\mathbb{E}[(\hat{\theta}_n(x) - \theta(x))^2]} < 1,$$

for x in suitable neighborhoods (depending on the fraction a) around the best projected split points of a regression tree (e.g. the parameter d^0 in (10) for a stump), and where $\theta(x) = \lim_{n\to\infty} \mathbb{E}[\hat{\theta}_n(x)]$. That is, subagging asymptotically reduces the MSE for x in neighborhoods around the unstable split points, a fact which we may also compare with Figure 2. Moreover, one can argue that globally,

$$\mathbb{E}[(\hat{\theta}_{n;SB(m)}(X) - \theta(X))^2] \overset{\text{approx.}}{<} \mathbb{E}[(\hat{\theta}_n(X) - \theta(X))^2]$$

for n large and where the expectations are taken also over (new) predictors X.

For subagging with small order $m = o(n)$, such a result is no longer true: the reason is that small order subagging will then be dominated by a large bias (while variance reduction is even better than for fraction subagging with $m = [an],\ 0 < a < 1$).

Similarly as for the toy example in section 2.1.1, subagging smoothes the hard decisions in a regression tree resulting in reduced variance and MSE.

2.3. Bagging basis functions in MARS

We discuss here the effect of bagging on the basic ingredient in MARS [9]. For a one-dimensional predictor variable, the basis function in MARS is a piecewise linear spline function $[x - d]_+ = (x - d)\mathbf{1}_{[d \le x]}$. Its estimated version takes the form

$$\hat{\theta}_n(x) = \hat{\beta}_n[x - \hat{d}_n]_+,\ x \in \mathbb{R}, \tag{11}$$

with the least squares estimates

$$(\hat{\beta}_n, \hat{d}_n) = \text{argmin}_{\beta,d} \sum_{i=1}^{n} (Y_i - \beta[X_i - d]_+)^2$$

for the best projected values $(\beta^0, d^0) = \text{argmin}_{\beta,d}\, \mathbb{E}[(Y - \beta[X - d]_+)^2]$. It is possible to

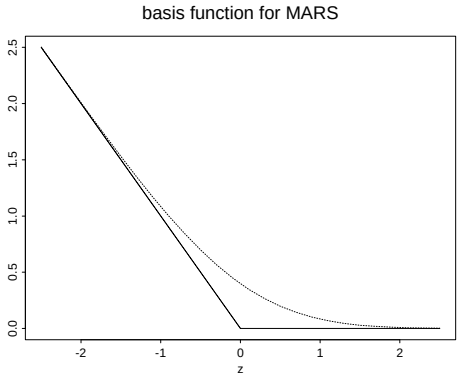

Figure 3. MARS basis function from (11) at $x = x_n(0) = d^0$ as in (12). Function $g(z)$ and $g_{Bag}(z)$ (dotted line) from (13), defining the asymptotics of $\hat{\theta}_n(x_n(0))$ and its bagged version, respectively.

derive the asymptotic behavior of this MARS basis function for x in a neighborhood of the best projected knot d^0, i.e. for

$$x = x_n(c) = d^0 + c\sigma_d n^{-1/2}, \tag{12}$$

where σ_d^2 is the asymptotic variance of $\hat{d}_n$. The smoothing effect of bagging with the MARS basis function can be described as follows, assuming some regularity conditions:

$$\begin{aligned} &n^{1/2}\sigma_d^{-1}\hat{\theta}_n(x_n(c)) \to_D g(Z) = \beta^0(c - Z)\mathbf{1}_{[Z \le c]}, \\ &n^{1/2}\sigma_d^{-1}\hat{\theta}_{n;Bag}(x_n(c)) \to_D g_{Bag}(Z) = \beta^0\{(c - Z)\Phi(c - Z) + \varphi(c - Z)\}, \end{aligned} \tag{13}$$

where $Z \sim \mathcal{N}(0, 1)$, see [4]. The functions $g(\cdot)$ and $g_{Bag}(\cdot)$ are displayed in Figure 3. We see that already the original MARS basis function estimator corresponds to a continuous limiting function $g(\cdot)$, in contrast to regression stumps in (9). Also, the smoothing effect of bagging, described by the asymptotic function $g_{Bag}(\cdot)$, turns out to have a small effect only. The MSEs for the MARS basis function estimator and its bagged version, which can be computed from (13), are displayed in Figure 4. We clearly see that the bagging improvement is at most only very marginal.

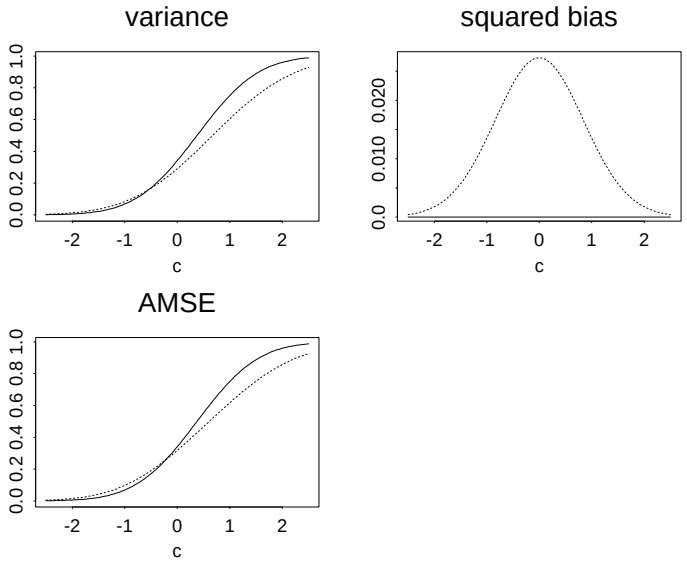

Figure 4. MARS estimator $\hat{\theta}_n(x_n(c))$ from (11) with $x_n(c)$ from (12). Asymptotic variance, squared bias and mean squared error (AMSE) (the target is $\theta(x) = \lim_{n\to\infty}\mathbb{E}[\hat{\theta}_n(x)]$), standardized by the factor $n\sigma_d^{-2}$, for the estimator $\hat{\theta}_n(x_n(c))$ in (11) (solid line) and for the bagged estimator $\hat{\theta}_{n;B}(x_n(c))$ (dotted line), as a function of c.

2.3.1. Effect of unstable variable selection

The analysis in section 2.3 only covers the case with one-dimensional predictor variables and one linear piecewise function. It can be viewed as an analogue for regression stumps

with one predictor. While for the latter, (su-)bagging does improve the MSE substantially (in theory and also in simulated finite sample cases [4]), we see no relevant asymptotic gain for bagging with one MARS basis function. With higher dimensional predictors or more basis functions per predictor, there could be *another* effect of bagging, besides the smoothing effect which is very dominant in regression trees, but negligible for spline functions in MARS. We will demonstrate empirically in section 3, that bagging can be very unstable for MARS (in terms of choosing basis functions) with more than one predictor variable and has a chance to perform very poorly; this is in sharp contrast to bagging decision trees which we have found to be *always* better in terms of MSE compared to original trees. But a version of bagging based on robust aggregation, see section 3.1, is found to be effective for MARS, yielding better MSE performance than the original MARS. This better performance is not due to bagging-type smoothing (at least not asymptotically as argued above); it rather seems to help reducing the negative effect of unstable variable selection in MARS.

3. Numerical results and Bragging

We show here the bagging procedure in action for synthetic data from three different models:

$$\begin{aligned}
&\text{(M1)} \quad X = (X^{(1)}, \ldots, X^{(10)}) \sim \text{Unif}([0,1]^{10}),\ \varepsilon \sim \mathcal{N}(0,6),\\
&\qquad Y = 10\sin(\pi X^{(1)} X^{(2)}) + 20(X^{(3)} - .5)^2 + 10X^{(4)} + 5X^{(5)} + \varepsilon.\\
&\text{(M2)} \quad X = (X^{(1)}, \ldots, X^{(5)}) \sim \mathcal{N}_5(0, I),\ \varepsilon \sim \mathcal{N}(0,4),\\
&\qquad Y = X^{(1)} + .5X^{(2)} + .8X^{(3)} + .5X^{(4)} + .3X^{(5)} + 2X^{(1)}X^{(2)} + 3X^{(2)}X^{(3)} + \varepsilon.\\
&\text{(M3)} \quad X^{(1)}, X^{(2)} \text{ i.i.d.} \sim \text{Unif}(\{0,1\}), X^{(3)}, X^{(4)} \text{ i.i.d.} \sim \text{Unif}(\{0,1,2,3\}),\\
&\qquad X^{(5)} \sim \text{Unif}(\{0,1,2,3,4,5,6,7\}),\\
&\qquad X^{(1)}, X^{(2)}, \ldots, X^{(5)} \text{ independent},\ \varepsilon \sim \mathcal{N}(0,4),\\
&\qquad Y = \mathbf{1}_{[X^{(1)}=0]} - 3\mathbf{1}_{[X^{(1)}=1]} + .5\mathbf{1}_{[X^{(2)}=0]} + 2\mathbf{1}_{[X^{(2)}=1]} + .8\mathbf{1}_{[X^{(3)}=0]} - 2\mathbf{1}_{[X^{(3)}=1]}\\
&\quad + \ 2\mathbf{1}_{X^{(3)}=2]} - 1\mathbf{1}_{[X^{(3)}=3]} + .5\mathbf{1}_{[X^{(4)}=0]} + 1.2\mathbf{1}_{[X^{(4)}=1]} - .9\mathbf{1}_{[X^{(4)}=2]} + 1.8\mathbf{1}_{[X^{(4)}=3]}\\
&\quad + \ .3\mathbf{1}_{[X^{(5)}=0]} - .6\mathbf{1}_{[X^{(5)}=1]} + .9\mathbf{1}_{[X^{(5)}=2]} - 1.2\mathbf{1}_{[X^{(5)}=3]} + 1.5\mathbf{1}_{[X^{(5)}=4]} - 1.8\mathbf{1}_{[X^{(5)}=5]}\\
&\quad + \ 2.1\mathbf{1}_{[X^{(5)}=6]} - 2.4\mathbf{1}_{[X^{(5)}=7]} + 2\mathbf{1}_{[X^{(1)}=0, X^{(2)}=1]} + 3\mathbf{1}_{[X^{(2)}=0, X^{(3)}=1]} + \varepsilon.
\end{aligned}$$

Sample size is always chosen as $n = 300$. Model (M1)also known as Friedman #1, has 5 ineffective predictor variables is; in (M2), all predictor variables are effective and some of them occur also as strong interactions; in (M3), all predictor variables are factors and effective, some of them also occurring as strong interactions. The signal to noise ratios $\text{Var}(f(X))/\text{Var}(\varepsilon)$, where $f(\cdot) = \mathbb{E}[Y|X = \cdot]$, are 3.97 for (M1), 3.77 for (M2) and 3.08 for (M3) : thus, all models have similar orders of signal to noise ratio. All our numerical results are based on 100 independent model simulations and we always use 100 bootstrap replications for bagging.

We report the MSE $\mathbb{E}[(\hat{\theta}(X) - f(X))^2]$, where the expectation is also over new predictors X; sometimes, we also consider the random variable of the (test set) squared error $(\hat{\theta}(X) - f(X))^2$.

In case of trees, we never lose with bagging in terms of MSE (and also in terms of the test set squared error; this cannot be seen from Table 1). The model (M3) with discrete

predictors (factors) is beyond what has been treated in theory. Nevertheless, we also see here that bagging "works well". Bagging MARS is highly unstable: the MSE is worse with bagging while the median squared error performance is better.

As pointed out in section 2.3, bagging a piecewise linear spline does not reduce the MSE asymptotically in case of a one-dimensional predictor variable. Also, it is expected that from an asymptotic point of view with fixed number of predictors $d < \infty$, the same predictor variables will be selected for the original and the bootstrap samples. However, the asymptotic stability of variable selection may be misleading for practical applications: when running MARS on a bootstrap sample, we will often see that the terms selected by MARS are different from the ones selected from the original sample. In particular, the chance that this happens increases when more terms in the MARS algorithm are allowed. Bagging MARS has a potential to reduce the test set squared error, due to averaging over unstable variable selection, for some individual datasets. But there is also a chance for extremely bad performance on some data: overall (on average), the MSE of MARS becomes large. But there is a simple trick to improve the highly unstable behavior of bagging MARS, as described next.

Table 1
Mean squared error (MSE) and quantiles of squared error in models (M1)-(M3) for various methods. "tree" indicates the default regression tree in `R` using the function `rpart`; "MARS (deg=2)" indicates MARS constraining the interaction terms to be at most of order 2 (function `mars` in `R` with "degree" parameter set to 2). Number of bootstrap replications in bagging is 100. Number of simulations is 100.

model & method	MSE	(0, 0.25, 0.5, 0.75, 1)-quantiles of squared error
(M1), tree	10.56	(8.66, 9.84, 10.52, 11.12, 14.22)
(M1), bagged tree	5.81	(4.55, 5.42, 5.68, 6.09, 7.22)
(M1), bragged tree	6.13	(4.86, 5.78, 6.07, 6.47, 7.77)
(M1), MARS (deg=2)	1.60	(0.77, 1.25, 1.47, 1.82, 5.71)
(M1), bagged MARS (deg=2)	1.83	(0.45, 0.68, 0.85, 1.05, 71.09)
(M1), bragged MARS (deg=2)	0.81	(0.41, 0.64, 0.79, 0.92, 1.55)
(M2), tree	11.30	(7.07, 8.67, 10.91, 13.14, 19.84)
(M2), bagged tree	7.61	(4.52, 6.48, 7.50, 8.56, 11.52)
(M2), bragged tree	7.32	(4.29, 5.95, 7.09, 8.26, 11.13)
(M2), MARS (deg=2)	4.40	(0.18, 0.69, 1.01, 1.30, 217.79)
(M2), bagged MARS (deg=2)	22.33	(0.27, 0.50, 0.67, 1.43, 1193.73)
(M2), bragged MARS (deg=2)	0.43	(0.18, 0.32, 0.39, 0.54, 1.35)
(M3), tree	3.04	(2.23, 2.75, 3.00, 3.33, 4.03)
(M3), bagged tree	1.70	(1.17, 1.53, 1.68, 1.82, 2.29)
(M3), bragged tree	1.90	(1.40, 1.69, 1.88, 2.05, 2.61)

3.1. Bragging

Bragging is a sobriquet for **b**ootstrap **r**obust **agg**regat**ing**. Instead of the sample mean in (1) in the bagging algorithm (see Definition 1), we use a robust location estimator for the realized bootstrapped estimators $\hat{\theta}_n^{*b}(\cdot),\ b = 1, \ldots B$. We propose to use

$$\hat{\theta}_{n;Brag} = \text{median}(\{\hat{\theta}_n^{*b}(\cdot); b = 1, \ldots, B\}).$$

We also looked at other robust location estimators for aggregating the bootstrapped estimates $\hat{\theta}_n^{*b}(\cdot)$'s such as Huber's estimator and Hampel's redescending M-estimator, but aggregation with the sample median was found to be slightly better.

We show the results for bragging in Table 1 as well. We see that bragging MARS brings a substantial improvement over bagging MARS and over original MARS. The bootstrapped MARS estimates can be very poor resulting in a bad bagged estimator; since aggregation with the sample median is highly robust, bragging MARS does not suffer from such instabilities. In case of trees, bagging is stable ("always" reducing MSE), and bragging and bagging have about the same performance.

3.1.1. Some further experiments with the Friedman #1 model

We consider a version of model (M1), including also dimensionality $d = 20$, and with $\text{Var}(\varepsilon) = 1$:

$$\begin{aligned}\text{(M4,d)} \qquad & X = (X^{(1)}, \ldots, X^{(d)}) \sim \text{Unif}([0,1]^d),\ d \in \{10, 20\},\ \varepsilon \sim \mathcal{N}(0,1), \\ & Y = 10 \sin(\pi X^{(1)} X^{(2)}) + 20(X^{(3)} - .5)^2 + 10X^{(4)} + 5X^{(5)} + \varepsilon.\end{aligned}$$

We consider three different sample sizes $n \in \{50, 300, 1000\}$. Particularly for $d = 20$ and $n = 50$, the data is very high-dimensional relative to sample size.

As basis algorithms, we consider regression trees, MARS (deg=1) which has no interaction terms (i.e. an additive model with forward variable selection) and MARS (deg=2) which allows up to second order interaction terms.

Figure 5 displays the performance of the MSEs for the basis algorithms and their bagged and bragged versions. We see that bagging or bragging trees pays-off even asymptotically: the explanation for this is given in section 2. For MARS, bagging or bragging becomes more ineffective for larger sample size: this is again consistent with theory (for $d < \infty$), because the selection of predictor variables is asymptotically stable and bagging is not inducing any smoothing effect (see section 2.3) which would help to reduce the MSE. For MARS allowing second order effects (deg=2), we need a larger sample size than $n = 1000$ to make the selection among the $d(d+1)/2$ terms stable (instead of the d terms when no interaction terms are allowed).

Our little demonstration should help to point out that from an asymptotic point of view, bagging or bragging MARS is not yielding much gain in terms of MSE. The reason is that bagging MARS is not inducing any relevant smoothing effect as for trees, which would be beneficial in terms of MSE. With trees, we exploit again empirically, that bagging is paying off even when n is large.

From a more practical point of view, when sample size is large, we would typically allow interaction terms of order 3 or 4 or higher (or more terms per predictor); this would then put us back to the case where predictor terms selection would be unstable. Thus, for many applications in practice, bragging MARS will be very useful; and bagging MARS

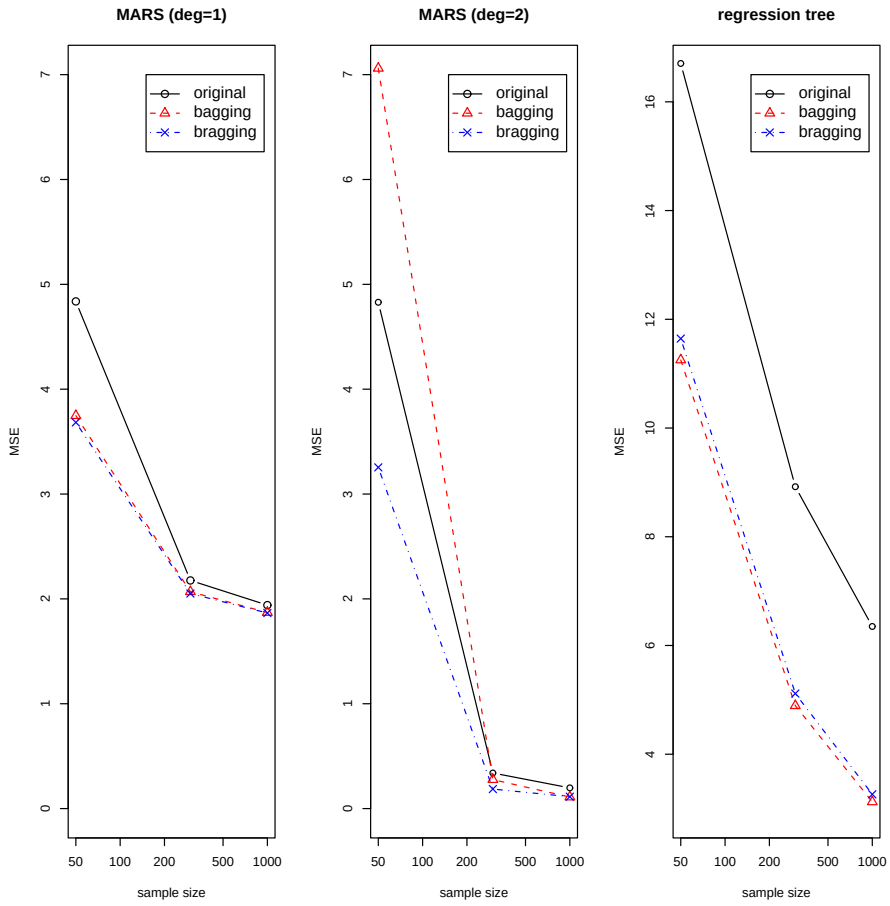

Figure 5. Mean squared error (MSE) in model (M4,d=10) with sample sizes $n \in \{50, 300, 1000\}$. MARS (deg=1) is MARS without interaction terms, MARS (deg=2) is MARS with interactions up to degree 2 (function `mars` in `R`), and regression tree is with at least 5 observations in every terminal node (function `rpart` in `R` with "minbucket" parameter set to 5).

can be dangerous as indicated in Table 1. To support the last point, we also looked at the MSE in model (M4,d=20) for sample size $n = 50$, i.e. for a very high-dimensional problem relative to sample size:

(M4,d=20)	MARS (deg=2)	bagged MARS (deg=2)	bragged MARS (deg=2)
MSE	7.10	57.95	4.64

4. BagBoosting and an application for tumor classification using microarray gene expression data with very large d

BagBoosting is using bagging as a "module" in a boosting algorithm. Boosting, originally proposed by Freund and Schapire [8] is an algorithm which builds a linear combi-

nation of estimators,

$$F_m(\cdot) = \sum_{j=1}^{m} \alpha_j \hat{\theta}_{j;X,U}(\cdot), m = 1, 2, \ldots,$$

where $\alpha_j > 0$ are adaptively chosen weights and $\hat{\theta}_{j;X,U}(\cdot)$ are function estimates based on current generalized residuals $U_1, \ldots, U_n$ and the available predictors $(X_1, \ldots, X_n)$. In boosting terminology, the function estimator is called a "learner". For example, the learner could be a regression tree, yielding function estimates $\hat{\theta}_{j;X,U}(\cdot)$ based on the predictor variables and current generalized residuals (X_i, U_i), $i = 1, \ldots, n$. The linear combination weights α_j are from a one-dimensional line search in a gradient descent algorithm, and the number of iterations m is a tuning parameter of boosting, describing when to stop the algorithm. For more details, see for example [11].

BagBoosting. BagBoosting, proposed in Bühlmann and Yu [3], is a boosting algorithm where the learner is a bagged function estimator. For example, the learner could be a bagged stump, i.e. a bagged two-node regression tree.

Our BagBoosting algorithm is different from stochastic gradient descent (Friedman [10]) who uses a regression tree learner based on *one* subsample, randomly chosen in every boosting iteration, whereas BagBoosting uses an aggregated (su-)bagged tree in every boosting iteration.

We consider here a dataset consisting of microarray gene expression levels of 40 tumor and 22 normal colon tissues: for each tissue 6'500 human genes are measured using the Affymetrix technology. A selection of 2'000 genes with highest minimal intensity across the samples has been made, and these data are publicly available at `http://microarray.princeton.edu/oncology`. We pre-processed the data by carrying out a base 10 logarithmic transformation and standardizing each tissue sample to zero mean and unit variance across the gene expressions. Summarizing, we have data (X_i, Y_i) with $Y_i \in \{0, 1\}$ (cancerous and normal type) and $X_i \in \mathbb{R}^{2000}$ (2'000 gene expression levels) with $\sum_{j=1}^{2000} X_i^{(j)} = 0$, $\sum_{j=1}^{2000} (X_i^{(j)})^2/1999 = 1$ $(i = 1, \ldots, n = 62)$.

The entire classification method is then a two-step scheme where first the 200 most significant genes are selected according to a two-sample Wilcoxon test, followed by LogitBoost (Friedman et al. [11]) with stumps, or BagBoosting using LogitBoost with bagged stumps, or a (single) classification tree (these three latter methods are run on the 200 selected genes only). We emphasize that the selection of the 200 genes is part of the classification method and based on training data only (in cross-validation experiments). The theory in section 2 tells us that a bagged stump is better (in terms of MSE) than a stump, and it is then heuristically clear that BagBoosting with a bagged stump is better than boosting with original stumps. In the context of microarray gene expression data, this has been generally confirmed in [7]. Figure 6 exhibits an advantage of BagBoosting over LogitBoost, and both are much better than a single classification tree. Our cross-validated misclassification rates become even smaller when using leave-one-out cross-validation and are then very competitive with other published results [7]; but we think that the 2/3 training 1/3 test set cross-validation yields more reliable estimates for the unknown misclassification error.

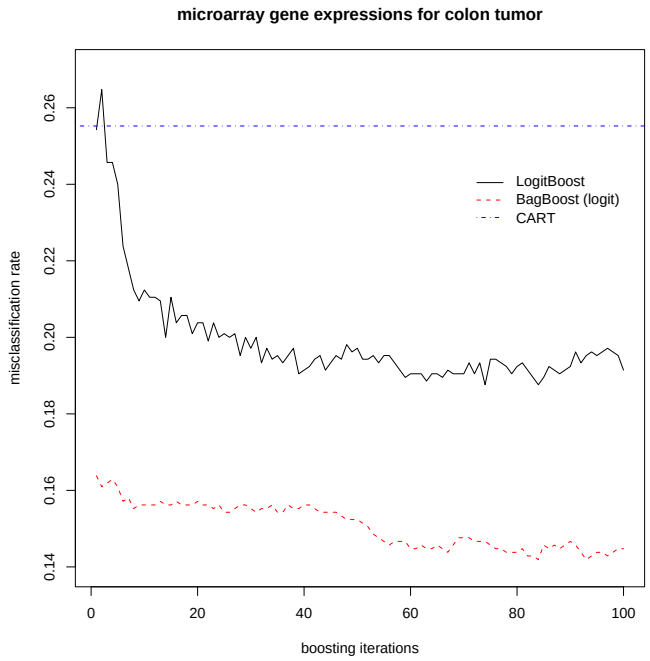

Figure 6. Misclassification errors for classifying cancerous versus normal colon tissue from 200 pre-selected microarray gene expressions, based on cross-validation with 50 random divisions into 2/3 training and 1/3 test sets. LogitBoost with stumps, BagBoosting is LogitBoost with bagged stumps, and classification tree with default settings for function `rpart` from `R`.

5. Conclusions

The theory and heuristics in section 2 describes that bagging is a smoothing operation in cases where the original estimator involves hard-decision indicator functions. The smoothing implies a variance and MSE reduction. The same applies also to subagging which is based on subsampling m data-points instead of resampling n points: the choice $m = [n/2]$ is in many cases almost equivalent to bagging, see also [5].

It has been empirically demonstrated in [4] that smoothing due to bagging and its corresponding variance and MSE reduction can be well seen in finite sample problems, even in low-dimensional settings with stumps for one predictor variable; see also section 3. It has been long "believed" that bagging would only help for high-dimensional, complex algorithms.

For smoother estimators than decision trees, bagging doesn't yield a first order smoothing effect: Buja and Stuetzle [5] have made this rigorous by showing that bagging a U-statistic has only second-order asymptotic effects on the MSE and bagging a U-statistics sometimes even increases MSE compared to the original U-statistics. A similar behavior also applies for linear spline functions which are used in MARS, as described in section 2.3: bagging MARS is likely not to be as effective as bagging decision trees and can

sometimes even result in very poor performance, as empirically shown in section 3.

Bragging based on robust aggregation, which is a new version of bagging proposed here, seems to have the ability to achieve another effect than smoothing. We showed empirically that bragging MARS often improves upon the original MARS algorithm. The probable reason which renders bragging MARS successful is that it averages (in a robust way) the instabilities of selected terms in MARS which then helps for variance reduction.

REFERENCES

1. Breiman, L. (1996). Bagging predictors. Machine Learning **24**, 123–140.
2. Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont (CA).
3. Bühlmann, P. and Yu, B. (2000). Discussion on Additive logistic regression: a statistical view of boosting, auths. J. Friedman, T. Hastie and R. Tibshirani. Annals of Statistics **28**, 377–386.
4. Bühlmann, P. and Yu, B. (2002). Analyzing bagging. Annals of Statistics **30**, 927–961.
5. Buja, A. and Stuetzle, W. (2002). Observations on bagging. Preprint. Available from http://ljsavage.wharton.upenn.edu/∼buja/
6. Chen, S.X. and Hall, P. (2003). Effects of bagging and bias correction on estimators defined by estimating equations. To appear in Statistica Sinica.
7. Dettling, M. (2003). BagBoosting for tumor classification with gene expression data. In preparation.
8. Freund, Y. and Schapire, R.E. (1996). Experiments with a new boosting algorithm. In Machine Learning: Proc. Thirteenth International Conference, pp. 148–156. Morgan Kauffman, San Francisco.
9. Friedman, J.H. (1991). Multivariate adaptive regression splines (with Discussion). Annals of Statistics **19**, 1–141 (with discussion).
10. Friedman, J.H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis **38**, 367–378.
11. Friedman, J.H., Hastie, T. and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. Annals of Statistics **28**, 337–407 (with discussion).
12. Politis, D.N., Romano, J.P. and Wolf, M. (1999). *Subsampling*. Springer, New York.