Published in final edited form as:
J Multivar Anal. 2008 March ; 25(2): 260–266.

Cross-Validated Bagged Learning

Maya L. Petersen[a], **Annette M. Molinaro**[b], **Sandra E. Sinisi**[a], and **Mark J. van der Laan**[a*]

[a]*Division of Biostatistics, University of California, Berkeley, School of Public Health, Earl Warren Hall 7360 Berkeley, California 94720-7360, phone: 510.642.3241 fax: 510.643.5163*

[b]*School of Public Health, Yale University, 60 College Street, New Haven, CT 06520-8034, phone: 203.785.4157 fax: 203.785.6912*

Abstract

Many applications aim to learn a high dimensional parameter of a data generating distribution based on a sample of independent and identically distributed observations. For example, the goal might be to estimate the conditional mean of an outcome given a list of input variables. In this prediction context, bootstrap aggregating (bagging) has been introduced as a method to reduce the variance of a given estimator at little cost to bias. Bagging involves applying an estimator to multiple bootstrap samples, and averaging the result across bootstrap samples. In order to address the curse of dimensionality, a common practice has been to apply bagging to estimators which themselves use cross-validation, thereby using cross-validation within a bootstrap sample to select fine-tuning parameters trading off bias and variance of the bootstrap sample-specific candidate estimators. In this article we point out that in order to achieve the correct bias variance trade-off for the parameter of interest, one should apply the cross-validation selector *externally* to candidate bagged estimators indexed by these fine-tuning parameters. We use three simulations to compare the new cross-validated bagging method with bagging of cross-validated estimators and bagging of non-cross-validated estimators.

Keywords

bootstrap aggregation; data-adaptive regression; resistant HIV; Deletion/Substitution/Addition algorithm

1 INTRODUCTION AND MOTIVATION

Many applications aim to use a learning data set from a particular data generating distribution to construct a predictor of an outcome as a function of a collection of input variables. One can define an optimal predictor as a parameter of the data generating distribution by defining it as the function of input variables which minimizes the expectation of a particular loss function (of the experimental unit and the candidate regression) with respect to the true data generating distribution. If one selects the squared error loss function (i.e., the square of the difference between the outcome and predicted value), then this optimal predictor is the conditional mean of the outcome, given the input variables. In the statistical literature such a location parameter

*Mark J. van der Laan *Email address:* laan@stat.berkeley.edu (Mark J. van der Laan). *URL:* http://www.stat.berkeley.edu/~laan (Mark J. van der Laan).

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

of the conditional distribution of the outcome given the input variables is often referred to as a regression.

In many applications the number of input variables is very large. As a consequence, assuming a fully parameterized regression model and minimizing the empirical mean of the loss function is likely to yield poor estimators. For example, assuming a linear regression model with all main terms and minimizing the the sum of squared residual errors is likely to yield a poor estimator since the number of main terms will typically be too large (resulting in over-fitting), and other functional forms of the input variables should be considered. In other words, many current applications demand nonparametric regression estimators. Because of the curse of dimensionality, minimizing the empirical mean of the loss function, i.e., the empirical risk, over all allowed regression functions results in an over-fit of the data.

As a consequence, many estimators follow the sieve loss-based estimation strategy. That is, a sequence of subspaces indexed by fine-tuning parameters is selected, the empirical risk over each subspace is minimized or locally minimized to obtain a subspace-specific (minimum empirical risk) estimator, and the fine-tuning parameter (i.e., the subspace) is selected using an appropriate method to trade off between bias and variance. Examples of fine- tuning parameters indexing constraints on the space of regression functions include initial dimension reduction, the number of terms in the regression model, and the complexity of the allowed functional forms (e.g., basis functions). Each specification of the fine-tuning parameters corresponds to a candidate estimator of the true underlying regression. In order to select among these candidate estimators (i.e., to select these fine-tuning parameters), most algorithms either minimize a penalized empirical risk or minimize the cross-validated risk.

Application of such “machine learning” algorithms to a data set commonly results in a very low dimensional fit. For example, in a recent HIV data application involving prediction of viral replication capacity based on the mutation profile of the virus, in spite of the fact that the employed algorithm searched over a high dimensional space of regression functions, a linear regression with two main terms and a single interaction was selected [1]. Although such an estimator is based on a sensible trade off between bias and variance, the resulting fit is disappointing from two perspectives. First, in many applications the true regression is believed to be a function of almost all variables, with many variables making small contributions. Second, a practitioner often wishes to obtain a measure of importance for each variable considered, and such a low dimensional fit reflects zero importance for all variables that do not appear in the estimator. It has been common practice to address the second issue by reporting many of the fits the algorithm has searched over, and to summarize these different fits in a particular manner. Initially, we also followed this approach, but came to the conclusion that the statistical interpretation of such a summary measure is unclear.

Based on these concerns, the following statistical challenge can be formulated: construction of nonparametric regression estimators that 1) are high dimensional, so that the majority of variables contribute to the obtained regression, and 2) still correspond with a sensible trade-off between bias and variance (and thereby have good asymptotic convergence properties). In the current article, we address this challenge. In order to construct high dimensional learners (i.e., 1), we employ the existing machine learning method “bootstrap aggregating” or “bagging” (or “aggregate prediction”), as introduced by Breiman [2]. However, in order to establish 2), we will provide a fundamental improvement to the current practice of bagging by changing the way cross-validation enters into the methodology.

Breiman suggested bagging as a method to stabilize (and thereby improve upon) a given highly variable estimator. Specifically, given an estimator, Breiman defined a corresponding bagged estimator as the average across bootstrap samples of the bootstrap sample-specific estimators.

Since different bootstrap samples typically result in different regression fits, the bagged estimator is typically a very high dimensional regression. Two applications of bagging are provided in random forest and linear regression [3]. In random forests, the bagged regression estimator is defined as an average of bootstrap-specific classification and regression tree (CART) estimators [4], where in each bootstrap sample CART is applied without cross-validation to obtain a fine partitioning. In the linear regression context, Breiman [2] proposed a bagged estimator as the average of bootstrap-specific cross-validated regression estimators, such as a linear regression estimator using forward selection and cross-validation to data adaptively select the size of the model. To conclude, in the current literature on bagging one either aggregates over-fitted regression estimators or one aggregates cross-validated regression estimators.

We note that the latter type of cross-validation within a bootstrap sample provides the right trade-off between bias and variance for the single sample estimator applied to the bootstrap sample. However, since the bagging operation reduces the variance and increases bias, it will typically result in the wrong trade-off for the corresponding bootstrap aggregated estimators. In this article we propose a cross-validated bagged estimator which 1) acknowledges that each estimator indexed by fine-tuning parameters corresponds with a bagged estimator, and 2) uses (external) cross-validation to select among these candidate bagged estimators, and possibly between these estimators and additional (e.g.) non-bagged estimators. By including non-bagged estimators in the set of candidate estimators, this procedure data adaptively selects between bagged and non-bagged estimators, which can be useful in cases where it is unclear if bagging actually improves the prediction performance: see our overview of some of the bagging literature below.

In order to assess the performance of the proposed cross-validated bagged estimator, we propose the use of cross-validation again: that is, the cross-validated bagged estimator is applied to a learning sample, and its fit is evaluated on a test sample, across different splits of the data into learning and test samples. The latter type of procedure involves double cross-validation.

The organization of this article is as follows. After presenting a brief review of bagging in Section 2, in Section 3 we present our cross-validated bagged estimator, presented in the context of the general unified loss-based estimation framework as introduced in [5]. That is, our estimator applies to any parameter which can be represented as the minimizer over the parameter space of an expectation of a loss function of the experimental unit at a parameter value, where we allow the loss function to be indexed by an unknown nuisance parameter. In particular, this general framework allows us to define the cross-validated bagged estimator of a conditional density, conditional hazard, or conditional location parameter (e.g., mean, median), based on censored and uncensored data. For example, our framework includes prediction of a survival time when the survival time is subject to right censoring. In Section 4 we compare our proposed cross-validated bagged estimator with the bagged cross-validated and bagged non-cross-validated estimators proposed by Breiman.

2 BRIEF REVIEW OF BAGGING

Bagging, or bootstrap aggregating, was introduced by Breiman [2] as a tool for reducing the variance of a predictor. The general idea is to generate multiple versions of a predictor and then use these to get an aggregated predictor. The multiple predictors are obtained by using bootstrap replicates of the data, and bagging is meant to yield gains in accuracy. Whether or not bagging will improve accuracy is related to the stability of the procedure that constructs each predictor [2]. Breiman [6] studied instability and stated that k-nearest neighbor methods are *stable*, but that neural networks, classification and regression trees, and subset selection in linear regression were *unstable* methods. Breiman [2] found that bagging works well for unstable methods.

Several approaches have been offered to combine different classifiers [7,8,9]. In addition, the following modifications of bagging have been proposed: “nice” bagging [10], sub-bagging or sub-sample aggregating [11], and iterated bagging or de-biasing [12]. We provide a brief overview of the various properties of bagging that have been studied and the application of bagging to available algorithms in the literature.

Friedman and Hall [13] show that bagging reduces variability when applied to highly nonlinear estimators such as decision trees and neural networks, and can also reduce bias for certain types of estimators. Breiman [12] shows that iterated bagging is effective in reducing both bias and variance. Buja and Stuetzle [14] look at bagging statistical functionals and *U*-statistics and apply bagging to CART [4]. They find that in the case of bagging CART, both variance and bias can be reduced. Buhlmann and Yu [11] define the notion of instability and analyze how bagging reduces variance in hard decision problems. Because hard decisions create instability, bagging is helpful to smooth these out, yielding smaller variance and mean squared error. They also look at the bagging effect on piecewise linear spline functions in multivariate adaptive regression splines (MARS) [15] and find that bagging is unnecessary for MARS. Borra and Ciaccio [16] also apply bagging to MARS, as well as to project pursuit regression (PPR) and local learning based on recursive covering (DART), and note that in most cases bagging reduces the variability of these methods. Bagging has been viewed from its ability to reduce instability [11] and its success with nonlinear features of statistical methods [13,14]. Hall and Samworth [17] address how performance depends on re-sample size.

Skurichina and Duin [10] offer several conclusions about bagging for linear classifiers; these include that bagging is not necessarily a stabilizing technique where stabilization is defined for linear classifiers, that the number of boot-strap replicates should be limited [2,10], that the usefulness of bagging can be determined by the instability of the classifier, and that bagging improves the performance of a classifier when the classifier is unstable. For computational considerations, it is helpful to have a sense of how many bootstrap replicates are adequate. Breiman [2] looked at 10 to 100 replicates and suggested that fewer replicates are required when the outcome is numerical.

3 THE CROSS-VALIDATED BAGGED LEARNER

Suppose that one observes a sample of n i.i.d. observations on a random variable O with data generating distribution P_0, which is known to be an element of a model $\mathcal{M}$. Let $\psi_0 = \Psi(P_0)$ be the parameter of interest of the data generating distribution P_0. We assume that the true parameter (value) ψ_0 can be defined in terms of a *loss function*, $(O, \psi) \to L(O, \psi)$, as the minimizer of the expected loss, or *risk*. That is, $\psi_0 = \Psi(P_0) = \arg\min_{\psi \in \Psi} \int L(o,\psi) dP_0(o)$, where the minimum is taken over the parameter space $\mathbf{\Psi} \equiv \{\psi(P) : P \in \mathcal{M}\}$.

In regression with a continuous outcome, a common loss function is the squared error loss, $L(O = (Y, W), \psi) = (Y - \psi(W))^2$, corresponding to the conditional mean $\psi_0(W) = E_0[Y \mid W]$, and if $\psi_0 = dP_0/d\mu$ is the actual density of O or a sub-vector of O, then $L(O, \psi) = -\log \psi(O)$. As in the unified loss based estimation approach presented in [5], it is allowed that the loss function depends on a nuisance parameter $\Upsilon(P_0)$: that is $L(O, \psi) = L(O, \psi \mid \Upsilon\ (P_0))$. By allowing such unknown loss functions, this framework includes most parameters.

For the sake of notational convenience, we suppress the possible dependence of the loss function on a nuisance parameter in the notation, but we will point out at the appropriate places how this affects the proposed estimation procedure.

Let P_n denote the empirical probability distribution of the sample $O_1,\ldots,O_n$, which puts mass $1/n$ on each observation. Consider now a collection of candidate estimators $Pn \rightarrow \hat{\Psi}_s(P_n)$ indexed by a fine tuning parameter s ranging over a set A_n. For example, if $\psi_0(W) = E(Y \mid W)$, then $\hat{\Psi}_s(P_n)$ might represent a particular learning algorithm for estimation of $E(Y \mid W)$ indexed by fine tuning parameters s which are user supplied, such as a support-vector machine algorithm, a forward step-wise algorithm, logic regression [18], the D/S/A-polynomial regression algorithm [19], and so on. Another class of general examples is obtained by defining $\hat{\Psi}_s(P_n) \equiv \arg\ \min_{\psi \in \Psi_s} \Sigma_{i=1}^{n} L(O_i, \psi)$ as the minimizer of the empirical risk $\Sigma_i L(O_i, \psi)$ over a sub-parameter space $\mathbf{\Psi}_s \subset \mathbf{\Psi}$ indexed by s, given a collection of subspaces $\mathbf{\Psi}_s$, $s \in A_n$.

In the case that the loss function depends on a nuisance parameter $\Upsilon(P_0)$, one would estimate the nuisance parameter with an estimator $\hat{\Upsilon}(P_n)$, and minimize the empirical risk corresponding with the estimated loss function. Most estimators can be considered as approximate minimizers of the empirical risk, indexed by parameters defining the search algorithm, such as the space the algorithm searches over and the depth to which it searches the space. We note that we view the estimators as mappings $\hat{\Psi}_s$ from data, P_n, to the parameter space.

Given the empirical distribution P_n, let $P_n^{\#}$ denote the empirical distribution of a sample of n i.i.d. observations $O_1^{\#}, \ldots, O_n^{\#}$ from P_n. Given the s-specific estimator $\hat{\Psi}_s$ we can define a corresponding s-specific bagged estimator as $\tilde{\Psi}_s(P_n) \equiv E(\hat{\Psi}_s(P_n^{\#}) \mid P_n)$. To evaluate the conditional expectation, given the data P_n, one needs to draw many bootstrap samples $P_n^{\#}$ from the empirical probability distribution P_n. For each of these draws of size n, $P_{n1}^{\#}, \ldots, P_{nB}^{\#}$, the estimators $\hat{\Psi}_s(P_{nb}^{\#})$ based on the bootstrap sample $P_{nb}^{\#}$, $b = 1,\ldots,B$, are calculated and averaged: $\tilde{\Psi}_s(P_n) = \lim_{B \rightarrow \infty} \frac{1}{B} \Sigma_{b=1}^{B} \hat{\Psi}_s(P_{nb}^{\#})$.

This results in a set of candidate bagged estimators $\tilde{\Psi}_s(P_n)$ indexed by s. Our goal is to data-adaptively select the s which minimizes the risk of $\tilde{\Psi}_s(P_n)$ over A_n. For this purpose, we propose the *cross-validated bagged estimator* defined as: $\tilde{\Psi}(P_n) = \tilde{\Psi}_{\hat{S}(P_n)}(P_n)$ where $\hat{S}(P_n)$ is the cross-validation selector based on the loss function $L(\cdot,\cdot)$ corresponding to a cross-validation scheme defined by a random n dimensional vector $B_n \in \{0, 1\}^n$. A realization of $B_n = (B_n, 1,\ldots,B_{n,n})$ defines a particular split of the learning sample of n observations into a training set, $\{i \in \{1,\ldots,n\}: B_{n,i} = 0\}$, and a validation set, $\{i \in \{1,\ldots,n\}: B_{n,i} = 1\}$. We will denote the proportion of observations in the validation set with p. The empirical distributions of the training and validation sets are denoted by P_{n,B_n}^{0} and P_{n,B_n}^{1} respectively. Formally, the cross-validation selector $\hat{S}(P_n)$ (P_n) is defined as:

$$\begin{aligned} \hat{S}(P_n) &= \arg\ \min_{s \in A_n} E_{B_n} P_{n,B_n}^{1} L(\cdot, \tilde{\Psi}_s(P_{n,B_n}^{0})) \\ &= \arg\ \min_{s \in A_n} E_{B_n} \sum_{i,B_n(i)=1} L(O_i, \tilde{\Psi}_s(P_{n,B_n}^{0})). \end{aligned} \quad (1)$$

.

At the first equality we used the notation $Pf \equiv \int f(o)dP(o)$. If the loss function depends on an unknown nuisance parameter, then one estimates the unknown loss function on the training sample: that is, one replaces in (1) the loss function by $L(\cdot, \tilde{\Psi}_s(P_{n,B_n}^{0}) \mid \hat{\Upsilon}(P_{n,B_n}^{0}))$.

To calculate this selector of s, for each possible realization of the sample split B_n and index $s \in A_n$, B bootstrap samples $P_{n,B_n,b}^{0\#}$ of size $n(1 - p)$ are drawn from the training sample

$P^{0}_{n,B_n}, b = 1, \ldots, B$. For each of these *B* bootstrap samples we compute the corresponding *s*-specific estimators $\hat{\Psi}_s(P^{0\#}_{n,B_n,b})$ and average them to obtain: $\tilde{\Psi}_s(P^{0}_{n,B_n}) = \frac{1}{B}\Sigma_{b=1}^{B}\hat{\Psi}_s(P^{0\#}_{n,B_n,b})$. The empirical risk of this estimator over the validation sample can now be computed, and averaged over the different splits B_n, as in (1), which results in the so called cross-validated risk of the estimator $\tilde{\Psi}_s(P_n)$. The cross-validation selector is defined as the one which minimizes this cross-validated risk over $s \in A_n$. A schematic illustrating the implementation of the cross-validated bagged learner (applied using classification and regression trees, as employed in the simulations reported in Section 4, is shown in Figure 1.

3.1 Cross-Validated Bagged Learning versus Bagged Cross-Validated Learning

It is of interest to contrast the proposed cross-validated bagged estimator to the bagged cross-validated estimator as used in [2] and followed by other authors. In the bagged cross-validated approach, the selection of *s* via cross-validation is performed within each bootstrap sample. Subsequently the *B* bootstrap-specific estimators are averaged to arrive at the final estimator. Formally, within a bootstrap sample $P_n^{\#}$ the cross-validation selector of *s* can be defined as:

$$\hat{S}_{\mathrm{br}}(P_n^{\#}) = \arg\min_{s\in A_n} E_{B_n} P^{1,\#}_{n,B_n} L(\cdot, \hat{\Psi}_s(P^{0,\#}_{n,B_n})).$$

The corresponding estimator based on a single bootstrap sample $P_n^{\#}$ is thus defined as: $\hat{\Psi}_{\mathrm{CV}}(P_n^{\#}) = \hat{\Psi}_{\hat{S}_{\mathrm{br}}(P_n^{\#})}(P_n^{\#})$. Finally, the corresponding bagged estimator is the average over a large collection of bootstrap-specific estimators: $\tilde{\Psi}_{\mathrm{br}}(P_n) = E(\hat{\Psi}_{\mathrm{CV}}(P_n^{\#}) \mid P_n)$. Figure 1 illustrates implementation of the bagged cross-validated estimator, and contrasts it with the cross-validated bagged estimator, as well as with a single cross-validated estimator, and a bagged non-cross-validated estimator.

Using cross-validation within a bootstrap sample provides the right selection among the estimators $\hat{\Psi}_s(P_n^{\#}), s \in A_n$ regarding the trade-off between bias and variance. However, one would expect it not to perform the right trade-off between bias and variance for the actual bagged estimators $\tilde{\Psi}_s(P_n)$. The reason for this is that the bagged estimator should be less variable as a result of the averaging, and might be more biased due to the double sampling. An increase in bias is due to two (probably cumulative) sources: first, the bias introduced by applying an estimator to a bootstrap sample relative to the empirical sample; and second, the bias introduced by applying the estimator to the empirical sample relative to the truth. In general, the estimators $\hat{\Psi}_s, s \in A_n,$ and $\tilde{\Psi}_s, s \in A_n$, are very different classes of estimators, so that a good selector among the un-bagged estimators is not necessarily a good selector among the corresponding bagged estimators.

3.2 Performance of the Cross-Validation Selector

Let $d(\psi, \psi_0) = \int L(o,\psi)\, dP_0\, (o) - \int L(o, \psi_0) dP_0(o)$ denote the risk dissimilarity between a candidate ψ and the true ψ_0 implied by the loss function $L(\cdot,\cdot)$. The results on the cross-validation selector (see [20,21]) state that if the loss function is uniformly bounded in its arguments, then the difference of the risk dissimilarity of the cross-validated selected bagged estimator and the risk dissimilarity of the oracle selected bagged estimator is of the order log $K(n)/np$ plus possibly a term due to estimation of the nuisance parameter in the loss function. The oracle selected bagged estimator is defined as $\tilde{\Psi}_{\tilde{S}_{n(1-p)}(P_n)}(P_n), \text{ where } \tilde{S}_{n(1-p)}(P_n) = \arg\min_s E_{B_n}\int L(o, \tilde{\Psi}_s(P^{0}_{n,B_n}))dP_0(o)$. Thus for a given data set the oracle selector $\tilde{S}_{n(1-p)}(P_n)$ selects the bagged estimator (based on $n(1-p)$ observations) closest to the truth with respect to the risk dissimilarity.

These results only rely on the loss function to be uniformly bounded in the support of *O* and the parameter space. They imply that if the number of candidate estimators is polynomial in sample size (and, in the case that the loss function is unknown, that it can be estimated at a better rate than the convergence rate of the oracle selected estimator), then either the cross-validated selected estimator is asymptotically equivalent (up to the constant) to the oracle selected estimator, or it achieves the essentially parametric rate of convergence log *n/n*.

3.3 Cross-Validation Selection of the Degree of Bagging

In cases where there is a concern that the bagging operation might actually worsen the estimator it is a good idea to let cross-validation select between the original un-bagged estimator and the bagged estimator. In general, the following method might be of interest. Define $\tilde{\Psi}_{s,\alpha} = \alpha\tilde{\Psi}_s + (1-\alpha)\hat{\Psi}_s$ as the weighted average between the bagged and un-bagged estimator, $\alpha \in [0, 1]$, and use cross-validation to select (s, α). In this manner, the data are used decide to what degree α the bagging operation should be used, and, by our results establishing asymptotic equivalence with the oracle selector of (α, s), our cross-validated selected estimator will perform asymptotically at least as well as the non-bagged estimator and bagged estimator.

3.4 Assessing the Performance of the Cross-Validated Bagged Estimator

One can estimate the risk $\int L(o, \tilde{\Psi}(P_n))dP_0(o)$ of the cross-validated bagged estimator $\tilde{\Psi}(P_n)$ with the cross-validated risk of the estimator $\tilde{\Psi}(P_n) = \tilde{\Psi}_{\hat{S}(P_n)}(P_n)$ using a cross-validation scheme defined by a random vector $B_n^* \in \{0,1\}^n : E_{B_n^*}\Sigma_{i,B_n^*(i)=1}L(O_i, \tilde{\Psi}(P^0_{n,B_n^*}))$. This procedure would require carrying out a B_n-specific cross-validation scheme within each learning sample P^0_{n,B_n^*}, which is often referred to as double cross-validation.

4 Simulations

In this section we illustrate the proposed cross-validated bagged estimation methodology in comparison to both the bagged cross-validated and bagged non-cross-validated estimators suggested by Breiman, as well as in comparison to the non-bagged estimator. The implementation of each of the estimators is illustrated schematically in Figure 1. An extended description of these and additional simulation results are provided in an on-line technical report [22].

The models chosen for these simulations are the same as those implemented in both [2] and [15]. For each simulation, we evaluated four estimators: bagged cross-validated, bagged non-cross-validated, cross-validated bagged, and a single CART tree. For each, the same learning set P_n of size 200 was used for building and choosing the best predictor and the same independent test set of size 1000 was used to evaluate the fit and reflects the empirical risk estimated in the corresponding tables.

For the single tree, P_n was split into a training set P_n^0 and validation set P_n^1 and the training set was used to fit trees $\hat{\Psi}_s(P_n^0)$, where $s = (1,\ldots, S(n)$, *minbucket* = 7, *cp* = .01). The best level of the tree was chosen via 10-fold cross-validation based on the validation set, and the learning set was then used to fit a tree of that level, giving a final cross-validated tree estimator $\hat{\Psi}_{\text{CV}}(P_n)$.

For the bagged cross-validated estimator, the learning set P_n was used to generate $B = 100$ bootstrap samples and build B bootstrap-specific estimators $\hat{\Psi}_{\text{CV}}(P_n^{\#})$, where 10-fold cross validation was implemented to select the best tree level $\hat{S}_{\mathrm{br}}(P_n^{\#})$ for each bootstrap sample. The final estimator $\tilde{\Psi}_{\mathrm{br}_{\mathrm{CV}}}(P_n)$ was the average of the B bootstrap-specific cross-validated estimators.

For the bagged non-cross-validated estimator, the learning set P_n was used to generate $B = 100$ bootstrap samples and build B bootstrap-specific estimators $\hat{\Psi}_{S(n)}(P_n^{\#})$, each of maximum tree size $S(n)$. The final estimator $\hat{\Psi}_{\mathrm{br}_{\mathrm{FT}}}(P_n)$ was the average of the B bootstrap-specific estimators.

For our proposed cross-validated bagging scheme, P_n was split into a training set P_n^0 and validation set P_n^1 and the training set was used to generate $B = 100$ bootstrap samples. Bootstrap-specific trees $\hat{\Psi}_s(P_n^{0,\#})$ of each level $s = 1,\ldots, S(n)$ were averaged across bootstrap samples, yielding a vector of bagged estimators based on the training set and indexed by $s \quad (\tilde{\Psi}_s(P_n^0))$. Ten-fold cross-validation was used to select the best tree level for the bagged estimator ($\hat{S}(P_n)$) based on the independent validation sample. The resulting cross-validated bagged estimator was then applied to the learning sample $\tilde{\Psi}_{\hat{S}(P_n)}(P_n) = \tilde{\Psi}(P_n)$.

This entire procedure was repeated 100 times and the empirical risk averaged over the repetitions. All simulations were performed in the statistical package R [23] using the recursive partitioning algorithm rpart [24] for CART and the mlbench package for data generation.

4.1 Simulation 1

The first simulated data set, as described in Breiman [2], has 10 independent predictor variables $x_1,\ldots,x_{10}$ each of which is uniformly distributed over (0, 1). The response is given by $y = 10 sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + error$, where the $error \sim N(0, 1)$. The results are shown in Table 1.

4.2 Simulation 2

In the second simulated data set, as described in Breiman [2], there are 4 independent predictor variables $x_1,\ldots,x_4$ each of which is uniformly distributed over different ranges: $0 \le x_1 \le 100$, $20 \le (x_2/2\pi) \le 280$, $0 \le x_3 \le 1$, $1 \le x_4 \le 11$. The response is given by $y = (x_1^2 + (x_2 x_3 - (1/x_2 x_4))^2)^{1/2} + \mathrm{error}$, where the $error \sim N(0, .62)$. The results are shown in Table 2.

4.3 Simulation 3

In the third simulated data set, as described in Breiman [2], there are 4 independent predictor variables $x_1,\ldots,x_4$ each of which is uniformly distributed over different ranges: $0 \le x_1 \le 100$, $20 \le (x_2/2\pi) \le 280$, $0 \le x_3 \le 1$, $1 \le x_4 \le 11$. The response is given by $y = \arctan\left(\frac{x_2 x_3 - (1/x_2 x_4)}{x_1}\right. + \mathrm{error}$, where the $error \sim N(0, .86)$. The results are shown in Table 3.

4.4 Summary

The risks of the estimators generated by each bagging approach (bagged cross-validated, cross-validated bagged, and bagged non-cross-validated) were compared with the risk of the single cross-validated tree. In all simulations, all three bagged estimators resulted in a decreased risk as compared to the single tree. The magnitude of this decrease provides a measure of performance on which the three bagged estimators can be compared. Interestingly, in every case, the cross-validated bagged approach proposed in this paper resulted in a greater decrease in risk than did the bagged cross-validated approach. This result agrees with the theoretical argument, presented in Section 3, that the cross-validated bagged approach appropriately trades off bias and variance for the bagged estimator.

The improvement in risk resulting from cross-validating the bagged estimator, as compared to simply using the bagged estimator with the maximum tree level, is less dramatic; however, cross-validation does appear to provide some small benefit in terms of decreased risk. In comparing the cross-validated and non-cross-validated bagged estimators, it is interesting to note that the fine tuning parameter (tree level) selected by cross-validation is based on bagged estimators fit using the training sample. Because the training sample is a subset of the learning sample, one expects that the estimator based on the training sample will be more biased than an equivalent estimator based on the entire learning sample (for which cross-validation was performed with a completely independent dataset). In contrast, the non-cross-validated estimator fit on the learning sample is generally expected to be over-fit. Thus, the truth is expected to lie somewhere between the two estimators. It is interesting to note that, even the current simulation settings where lack of cross-validation seems to have little cost in terms of prediction error, the cross-validated bagged estimator in no simulation performs worse than the non-cross-validated bagged estimator.

Table 1–Table 3 also present estimates of the variability of the cross-validated risk. While estimating this variability is straightforward in a simulation setting, estimation of the variability in risk estimate based on a single sample poses a challenge. While beyond the scope of the present article, we point out that Dudoit and van der Laan [25] derive the influence curve for the cross-validated risk, making possible first-order inference for the risk estimate.

5 DISCUSSION

In this article we developed a general class of statistical learning algorithms for parameters that can be defined as a minimizer over the parameter space of the population mean of a loss function. The proposed class combines 1) a given statistical learning algorithm indexed by fine tuning parameters, 2) bootstrap aggregation, and 3) cross-validation. The result is a high dimensional fit of the parameter of interest that corresponds with a sensible trade-off between bias and variance. In particular, we implemented such an algorithm for prediction based on classification and regression trees. The results support our claim that when cross-validation is combined with bagging, cross-validation should be performed at the level of the bagged estimator itself (cross-validated bagged estimation), rather than prior to bagging at the level of the bootstrap sample (bagged cross-validated estimation).

As discussed by previous studies, the real benefit of bagging occurs if the original statistical learning algorithms are local learning algorithms. The learning algorithm employed in our simulations, CART, is vary variable. In contrast, other learning algorithms rely more on extrapolation; for example, the Deletion/Substituion/Addition algorithm ([19]) fits the true regression using a linear combination of polynomial basis functions. The optimal tradeoff between local learning and global smoothing via extrapolation will depend on the true underlying data-generating distribution. In a recent paper, Sinisi et. al. introduce the concept of super learning [26], in which the data are used to decide (via cross-validation) between candidate learning algorithms or convex combinations of such learners. This concept can readily be applied to bagging, by creating convex combinations of global and local learning algorithms, bagging the resulting estimators, and selecting between the resulting candidate estimators (both bagged and un-bagged) using cross-validation.

6 ACKNOWLEDGEMENTS

Maya Petersen is supported by a Predoctoral Fellowship from the Howard Hughes Medical Institute. Annette Molinaro is supported by an NCI-funded career award (K22 CA123146). This work was partially supported by NIH RO1 grant GM071397.

References

1. Birkner, MD.; Sinisi, SE.; van der Laan, MJ. Multiple testing and data adaptive regression: An application to HIV-1 sequence data. Statistical Applications in Genetics and Molecular Biology. URL http://www.bepress.com/sagmb/vol4/iss1/art8
2. Breiman L. Bagging predictors. Machine Learning 1996;24(2):123–140.
3. Breiman L. Random forests. Machine Learning 2001;45:5–32.
4. Breiman, L.; Friedman, JH.; Olshen, RA.; Stone, CJ. Classification and Regression Trees. Wadsworth International Group; 1984. The Wadsworth Statistics/Probability series
5. van der Laan, MJ.; Dudoit, S. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples, Tech. Rep. 130. Berkeley: Division of Biostatistics, University of California; 2003 Nov.. URL www.bepress.com/ucbbiostat/paper130/
6. Breiman L. Heuristics of instability and stabilization in model selection. Annals of Statistics 1996;24:2350–2383.
7. LeBlanc M, Tibshirani R. Combining estimates in regression and classification. Journal of the American Statistical Association 1996;91:1641–1650.
8. Breiman L. Stacked regressions. Machine Learning 1996;24:49–64.
9. Hothorn T, Lausen B. Bundling classifiers by bagging trees. Computational Statistics and Data Analysis 2005;49(4):1068–1078.
10. Skurichina M, Duin RPW. Bagging for linear classifiers. Pattern Recognition 1998;31:909–930.
11. Bühlmann P, Yu B. Analyzing bagging. Annals of Statistics 2002;30:927–961.
12. Breiman L. Using iterated bagging to debias regressions. Machine Learning 2001;45:261–277.
13. Friedman, JH.; Hall, P. On bagging and nonlinear estimation. 2000. URL http://www-stat.stanford.edu/~jhf/ftp/bag.ps
14. Buja, A.; Stuetzle, W. Observations on bagging. 2002. URL http://www-stat.wharton.upenn.edu/~buja/PAPERS/paper-bag-wxs.pdf
15. Friedman JH. Multivariate adaptive regression splines. The Annals of Statistics 1991;19(1):1–141.discussion by A. R. Barron and X. Xiao
16. Borra S, Ciaccio AD. Improving nonparametric regression methods by bagging and boosting. Computational Statistics and Data Analysis 2002;38:407–420.
17. Hall P, Samworth RJ. Properties of bagged nearest-neighbour classifiers. Journal of the Royal Statistical Society: Series B 2005;67(3):363–379.
18. Ruczinski, I.; Kooperberg, C.; LeBlanc, M. Logic regression; Journal of Computational and Graphical Statistics. 2003. p. 475-511.URL www.biostat.jhsph.edu/~iruczins/publications/publications.html
19. Sinisi, SE.; van der Laan, MJ. Deletion/Substitution/Addition algorithm in learning with applications in genomics. Statistical Applications in Genetics and Molecular Biology. URL http://www.bepress.com/sagmb/vol3/iss1/art18
20. van der Laan, MJ.; Dudoit, S.; Keleş, S. Asymptotic optimality of likelihood based cross-validation, Tech. Rep. 125. Berkeley: Division of Biostatistics, University of California; 2003 Feb.. URL www.bepress.com/ucbbiostat/paper125/
21. van der Laan, MJ.; Dudoit, S.; van der Vaart, AW. The cross-validated adaptive epsilon-net estimator, Tech. Rep. 142. Berkeley: Division of Biostatistics, University of California; 2004 Feb. URL www.bepress.com/ucbbiostat/paper142/
22. Molinaro, AM.; van der Laan, MJ. Cross-validating and bagging partitioning algorithms with variable importance, Tech. Rep. 185. Berkeley: Division of Biostatistics, University of California; 2005 Nov.. URL www.bepress.com/ucbbiostat/paper185/

23. Ihaka R, Gentleman R. A language for data analysis and graphics. Journal of Computational and Graphical Statistics 1996;5:299–314.

24. Therneau, T.; Atkinson, E. An introduction to recursive partitioning using the rpart routines, Tech. Rep. 61. Rochester: Division of Biostatistics, Mayo Clinic; 1997 Nov.. URL www.mayo.edu/hsr/techrpt/61.pdf

25. Dudoit S, van der Laan M. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. Statistical Methodology 2005;2(2):131–154.

26. Sinisi SE, Polley EC, Petersen ML, Rhee SY, van der Laan MJ. Super Learning: An application to prediction of HIV-1 drug susceptibility. Statistical Applications in Genetics and Molecular Biology. In Press

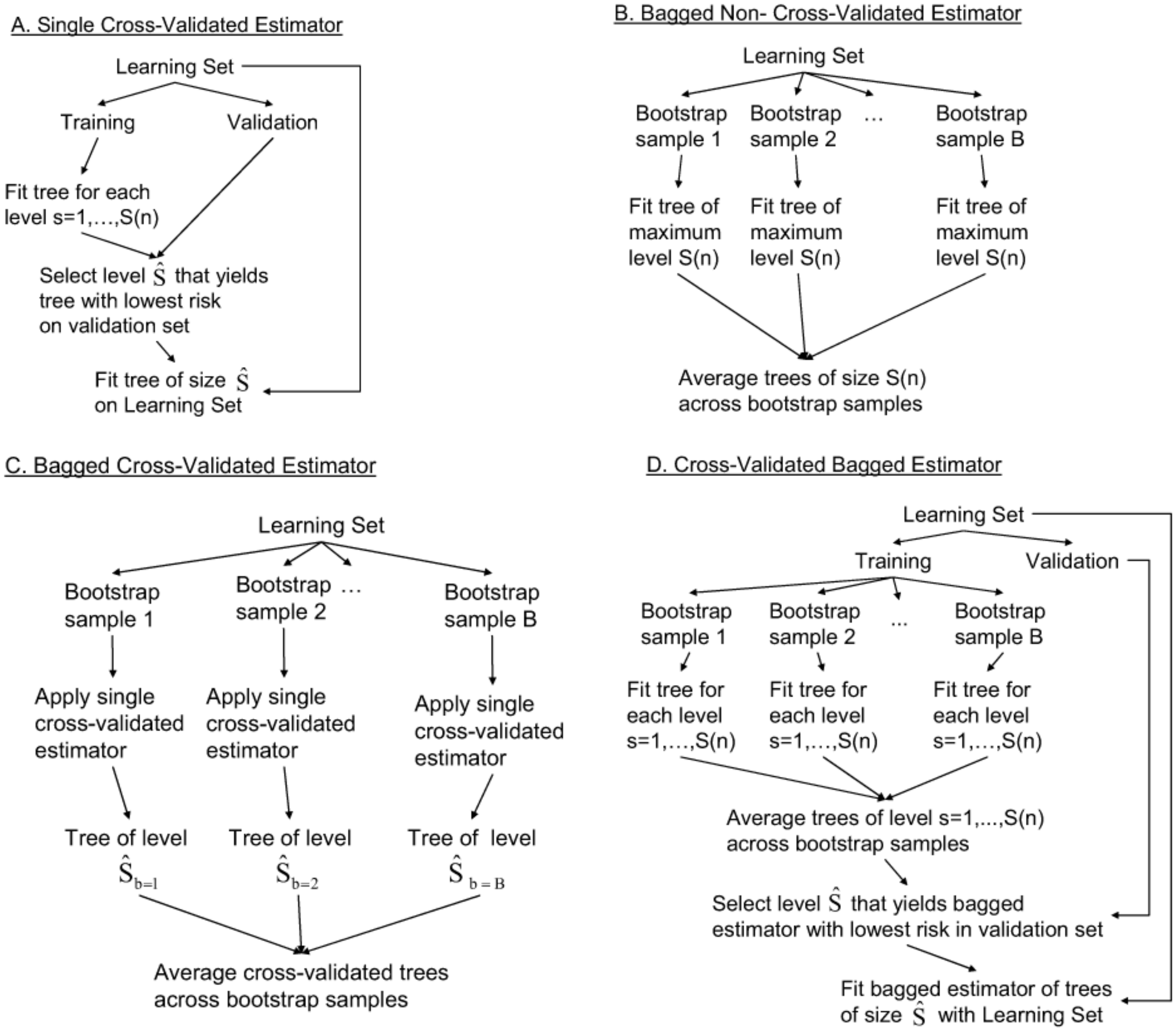
A. Single Cross-Validated Estimator
Learning Set
Training
Validation
Fit tree for each level s=1,…,S(n)
Select level Ŝ that yields tree with lowest risk on validation set
Fit tree of size Ŝ on Learning Set
B. Bagged Non- Cross-Validated Estimator
Learning Set
Bootstrap sample 1
Bootstrap sample 2
…
Bootstrap sample B
Fit tree of maximum level S(n)
Fit tree of maximum level S(n)
Fit tree of maximum level S(n)
Average trees of size S(n) across bootstrap samples
C. Bagged Cross-Validated Estimator
Learning Set
Bootstrap sample 1
Bootstrap sample 2
…
Bootstrap sample B
Apply single cross-validated estimator
Apply single cross-validated estimator
Apply single cross-validated estimator
Tree of level $\hat{S}_{b=1}$
Tree of level $\hat{S}_{b=2}$
Tree of level $\hat{S}_{b=B}$
Average cross-validated trees across bootstrap samples
D. Cross-Validated Bagged Estimator
Learning Set
Training
Validation
Bootstrap sample 1
Bootstrap sample 2
...
Bootstrap sample B
Fit tree for each level s=1,…,S(n)
Fit tree for each level s=1,…,S(n)
Fit tree for each level s=1,…,S(n)
Average trees of level s=1,...,S(n) across bootstrap samples
Select level Ŝ that yields bagged estimator with lowest risk in validation set
Fit bagged estimator of trees of size Ŝ with Learning Set

Figure 1.

Schematic contrasting alternative estimators, as applied in Simulations 1–3 using CART.

Table 1

Simulation 1

Estimator	Bootstrap Samples	Emp. Risk Mean	Emp. Risk Std.Dev.	% Improvement in Risk
Single Cross-Valid. tree $(\hat{\Psi}_{CV})$	0	20.83	3.07	
Bagged Cross-Valid. $(\tilde{\Psi}_{{br}_{CV}})$	100	15.63	1.37	25%
Bagged Non-Cross-Valid. $(\tilde{\Psi}_{{br}_{FT}})$	100	13.71	0.95	34%
Cross-valid. Bagged $(\tilde{\Psi})$	100	13.75	1.01	34%

Table 2

Simulation 2

Estimator	Bootstrap Samples	Emp. Risk Mean	Emp. Risk Std.Dev.	% Improvement in Risk
Single Cross-Valid. tree ($\hat{\Psi}_{CV}$)	0	15055.41	4709.78	
Bagged Cross-Valid. ($\tilde{\Psi}_{br_{CV}}$)	100	6283.21	1177.67	58%
Bagged Non-Cross-Valid. ($\tilde{\Psi}_{br_{FT}}$)	100	4883.19	978.73	67%
Cross-Valid. Bagged ($\tilde{\Psi}$)	100	4871.83	964.94	68%

Table 3

Simulation 3

Estimator	Bootstrap Samples	Emp. Risk Mean	Emp. Risk Std.Dev.	% Improvement in Risk
Single Cross-Valid. tree ($\hat{\Psi}_{CV}$)	0	0.8377	0.0365	
Bagged Cross-Valid. ($\tilde{\Psi}_{br_{CV}}$)	100	0.8005	0.0354	4%
Bagged Non-Cross-Valid. ($\tilde{\Psi}_{br_{FT}}$)	100	0.8217	0.0391	2%
Cross-Valid. Bagged ($\tilde{\Psi}$)	100	0.7951	0.0357	5%