Mach Learn (2009) 74: 235–255
DOI 10.1007/s10994-008-5096-0

Exact bootstrap k-nearest neighbor learners

Brian M. Steele

Received: 18 January 2008 / Revised: 7 August 2008 / Accepted: 5 December 2008 / Published online: 19 December 2008

Abstract Bootstrap aggregation, or bagging, is a method of reducing the prediction error of a statistical learner. The goal of bagging is to construct a new learner which is the expectation of the original learner with respect to the empirical distribution function. In nearly all cases, the expectation cannot be computed analytically, and bootstrap sampling is used to produce an approximation. The k-nearest neighbor learners are exceptions to this generalization, and exact bagging of many k-nearest neighbor learners is straightforward. This article presents computationally simple and fast formulae for exact bagging of k-nearest neighbor learners and extends exact bagging methods from the conventional bootstrap sampling (sampling n observations with replacement from a set of n observations) to bootstrap *sub*-sampling schemes (with and without replacement). In addition, a *partially* exact k-nearest neighbor regression learner is developed. The article also compares the prediction error associated with elementary and exact bagging k-nearest neighbor learners, and several other ensemble methods using a suite of publicly available data sets.

Keywords Bagging · k-nearest neighbor · Classification · Regression · Ensemble methods

1 Introduction

A statistical learner is a function that predicts an output variable Y from a concomitant input vector X. The learner is constructed from a training sample $\mathbf{Z} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ for which both inputs and outputs are observed. This article focuses on k-nearest neighbor learners (Cover and Hart 1967; Ripley 1996; Hastie et al. 2001), a collection of learners that are conceptually and computationally simple and often rival more sophisticated learners with respect to prediction error. In this article, the elementary k-nearest neighbor prediction of Y is defined as a mean over the k-nearest training sample inputs. Training observations are ordered according to distances between the target input X and the training sample inputs

Editor: Phil Long.

B.M. Steele (✉)
Dept. of Mathematical Sciences, University of Montana, Missoula, MT 59812, USA
e-mail: steeleb@mso.umt.edu

$X_1, \dots, X_n$. Classification problems are accommodated by defining Y to be a multinomial indicator of class membership. The k-nearest neighbor learner then is an estimator of the class membership probabilities and the k-nearest neighbor classification rule assigns an unlabeled observation to the class with the largest estimated probability of membership.

One approach to reducing prediction error in statistical learning problems is to combine many learners constructed from $\mathbf{Z}$ as a single *ensemble* learner. Notable examples are boosting (Friedman et al. 2000; Freund and Schapire 1997), stacking (Wolpert 1992), and random forests (Breiman 2001). This article is concerned with a particularly simple ensemble method called bootstrap aggregation, or *bagging* (Breiman 1996; Hastie et al. 2001; Skurichina and Duin 1998). The purpose of bagging is to construct a new learner that approximates the exact bootstrap expectation of the learner. In principle, the exact bootstrap expectation is computed with respect to the empirical distribution function, an estimate of the true underlying distribution function from which the sample has been drawn. Generally, the exact bootstrap expectation of the learner cannot be expressed analytically and a Monte Carlo algorithm is used to approximate the exact bootstrap expectation. Typically, this is accomplished by computing B predictions, each from a learner constructed from a bootstrap sample drawn randomly and with replacement from the training set. In the case of a quantitative output variable for example, the mean of the B predictions is an approximation of the exact bootstrap expectation and also the ensemble prediction of the output variable.

The application of bagging to k-nearest neighbor learners is unattractive from a computational standpoint because for each prediction, each of the B bootstrap samples must be ordered anew. Fortuitously, and unlike almost all other statistical learners, analytic formulae for computing the exact bootstrap expectations of the k-nearest neighbor learners are available (Caprile et al. 2004; Steele et al. 2003). Yet exact bagging methods for k-nearest neighbor learners have not been utilized in practical applications or studied in detail, presumably because the analytic formulae are complicated and computationally expensive in application. The purpose of this article is to introduce computationally simple and fast formulae for exact bagging of k-nearest neighbor learners. Additionally, two other advances involving k-nearest neighbor learners are presented. The first advancement extends exact bagging methods from conventional bootstrap sampling (sampling n observations with replacement from a set of n observations) to bootstrap *sub*-sampling schemes (with and without replacement), and the second advancement is the development of a *partially* exact k-nearest neighbor regression learner. This article concludes by comparing prediction error estimates among elementary and exact bagging k-nearest neighbor learners and several other ensemble methods using a suite of publicly available data sets.

2 Notation and terminology

Let $\mathcal{P}$ denote a population and $\mathbf{Z}$ denote a sample of n observations drawn from $\mathcal{P}$. An element of $\mathcal{P}$ is a pair $Z = (X, Y)$ consisting of an input vector $X = (x_1, x_2, \dots, x_p)$ and a output vector $Y = (y_1, y_2, \dots, y_c)$. It is assumed that each of the p input variables can be ordered so that the distance between any two input vectors can be determined. In the examples below for instance, all variables are quantitative and Manhattan distance was used throughout because of its simplicity; other metrics might have been used with little practical difference. Let $X_{1:n} = (X_{[1]}, X_{[2]}, \dots, X_{[n]})$ denote an ordered arrangement of the sample input vectors $X_1, X_2, \dots, X_n$ where the order is determined by the distances between X and $X_1, X_2, \dots, X_n$. The ordering $X_{1:n}$ *induces* an ordering on the data set $\mathbf{Z}$ (Bhattacharya 1974) which is denoted herein by $Z_{1:n} = (Z_{[1]}, Z_{[2]}, \dots, Z_{[n]})$, where $Z_{[i]} = (X_{[i]}, Y_{[i]})$. The induced ordering of the sample outputs is denoted as $Y_{1:n} = (Y_{[1]}, Y_{[2]}, \dots, Y_{[n]})$.

The k-nearest neighbor learner is developed for two classes of output variables. The first class are quantitative and scalar outputs. In this case, the elementary k-nearest neighbor learner is the linear combination

$$\eta(X|\mathbf{Z}) = w^T Y_{1:n}$$

where $w_i = k^{-1}$ if $i \le k$ and $w_i = 0$ if $k < i \le n$. The second class of output variables are multinomial variables arising in classification problems. In this situation, $\mathcal{P}$ is comprised of c disjoint classes $\mathcal{P}_1, \dots, \mathcal{P}_c$ and the output Y is a c-vector identifying class membership of Z. The jth element of Y is

$$y_j = \begin{cases} 1, & \text{if } Z \in \mathcal{P}_j, \\ 0, & \text{if } Z \notin \mathcal{P}_j. \end{cases}$$

The posterior probability of membership in class $\mathcal{P}_j$ is then $\pi_j = \Pr(Z \in \mathcal{P}_j|X) = \mathrm{E}(y_j|X)$. Herein, the k-nearest neighbor learner $\eta(X|\mathbf{Z})$ is an estimator of the posterior probability vector $\pi(X) = [\pi_1(X), \dots, \pi_c(X)]$. A compact expression for $\eta(X|\mathbf{Z})$ is developed by forming the matrix

$$\underset{n\times c}{\mathbf{Y}_{1:n}} = \begin{pmatrix} y_{[1],1} & \cdots & y_{[1],c} \\ \vdots & & \vdots \\ y_{[n],1} & \cdots & y_{[n],c} \end{pmatrix}, \tag{1}$$

where the ith row of $\mathbf{Y}_{1:n}$ is the c-vector $Y_{[i]}$. Then, $\eta(X|\mathbf{Z}) = w^T\mathbf{Y}_{1:n} = \widehat{\pi}(X|\mathbf{Z})$ is the k-nearest neighbor estimator of $\pi(X)$ and $\widehat{\pi}_j(X|\mathbf{Z})$ is the proportion of the k-nearest neighbors of Z belonging to class $\mathcal{P}_j$. The usual objective is to predict the class membership of Z from X, and so the k-nearest neighbor *classifier* is $\arg\max_j \widehat{\pi}_j(X|\mathbf{Z})$. In case of a tie among the largest values of $\widehat{\pi}_1(X|\mathbf{Z}), \dots, \widehat{\pi}_c(X|\mathbf{Z})$, the neighborhood size k may be successively increased until the tie is broken. Alternatively, the class prediction may be randomly selected from among the tied classes. In the examples below, ties were broken by increasing the neighborhood size.

3 Bootstrap aggregation

Bootstrap aggregation, or bagging (Breiman 1996; Hastie et al. 2001, Chap. 8; Hall and Samworth 2005) is an ensemble method of reducing the prediction error of a learner. Bootstrap aggregation is carried out by drawing B bootstrap samples from the training sample $\mathbf{Z}$, constructing a new learner from each, and averaging the predictions. If B bootstrap samples $\mathbf{Z}^{*1}, \dots, \mathbf{Z}^{*B}$ are drawn and used to construct learners $\eta(X|\mathbf{Z}^{*1}), \dots, \eta(X|\mathbf{Z}^{*B})$, then the bagged estimator of Y is

$$\eta^{*B}(X|\mathbf{Z}) = B^{-1}\sum_{b=1}^{B}\eta(X|\mathbf{Z}^{*b}). \tag{2}$$

Let F_n denote the empirical distribution function of $\mathbf{Z}$ placing probability mass n^{-1} at each $Z_i \in \mathbf{Z}$ and 0 elsewhere, and let $\mathbf{Z}^*$ denote a random sample of n observations drawn with replacement from F_n. The *exact* bagging learner $\eta^*(\cdot|\mathbf{Z})$ is the expectation of $\eta(\cdot|\mathbf{Z}^*)$ over F_n. The exact bootstrap expectation of a prediction can be expressed as

$$\eta^*(X|\mathbf{Z}) = \mathrm{E}[\eta(X|\mathbf{Z}^*)|F_n] = n^{-n}\sum_{i\in\mathcal{I}}\eta(X|\mathbf{Z}^i) \tag{3}$$

where $\mathcal{I}$ is the set of all n-tuples formed by choosing n integers with replacement from $\{1,\ldots,n\}$ and $\mathbf{Z}^i = (Z_{i_1},\ldots,Z_{i_n})$ is an n-tuple of elements drawn from $\mathbf{Z}$. As the number of elements in $\mathcal{I}$ is very large, it is generally not feasible to compute the exact bootstrap expectation of a statistical learner. For this reason, the bagged learner given in formula (2) is commonly used as an estimator of the exact bootstrap expectation. An important exception to the general intractability of the exact bootstrap expectation are k-nearest neighbor learners constructed with weights w_i not depending on Z_i for $1 \le i \le n$. Remark: an example of weights that *do* depend on Z_i are weights computed from the distances between the target and the neighbors of the target. For the remainder of this article, w_i is a weight not depending on Z_i. The next section develops analytic formulae for the exact bootstrap expectation of a k-nearest neighbor learner.

3.1 The exact bootstrap expectation of the k-nearest neighbor learner

Consider first prediction of a scalar output y associated with a population unit $Z = (y, X)$. Given a bootstrap sample $\mathbf{Z}^* = \{(y_1^*, X_1^*),\ldots,(y_n^*, X_n^*)\}$ and an input vector X, the distances between X and $X_i^*, i = 1,\ldots,n$ induce an ordering $Z_{[1]}^*,\ldots,Z_{[n]}^*$ on $\mathbf{Z}^*$. Concurrently, the ordering on the X_i^*'s induces the bootstrap order statistic $Y_{1:n}^* = (y_{[1]}^*,\ldots,y_{[n]}^*)$. The exact bootstrap expectation of a k-nearest neighbor learner $\eta(X|\mathbf{Z}) = w^T Y_{1:n}$ is

$$\eta^*(X|\mathbf{Z}) = \mathrm{E}(w^T Y_{1:n}^*|F_n) = \sum_{i=1}^{n} w_i \mathrm{E}(y_{[i]}^*|F_n). \tag{4}$$

The only possible realization of $y_{[i]}^*$ is one of $y_{[1]},\ldots,y_{[n]}$, and $y_{[i]}^*$ will be $y_{[j]}$ if and only if $X_{[i]}^* = X_{[j]}$ and equivalently, $Z_{[i]}^* = Z_{[j]}$. The bootstrap expectation of $y_{[i]}^*$ is thus

$$\mathrm{E}(y_{[i]}^*|F_n) = \sum_{j=1}^{n} \Pr(Z_{[i]}^* = Z_{[j]}|F_n) y_{[j]}.$$

An analytic formula for computing the *bootstrap probability* $\Pr(Z_{[i]}^* = Z_{[j]}|F_n)$ is obtained by noting that the number of elements in $\mathbf{Z}^*$ drawn from the set of j nearest observations $\{Z_{[1]},\ldots,Z_{[j]}\}$ is a binomial random variable with parameters n and j/n because the elements of the bootstrap sample are drawn independently and with replacement from $\{Z_{[1]},\ldots,Z_{[n]}\}$. Let $S_j^* \sim \mathrm{Bin}(n, j/n)$ denote the number of elements in $\mathbf{Z}^*$ drawn from $\{Z_{[1]},\ldots,Z_{[j]}\}$. The event $Z_{[i]}^* = Z_{[j]}$ will occur if and only if at least i observations are sampled from $Z_{[1]},\ldots,Z_{[j]}$ and less than i elements are sampled from $Z_{[1]},\ldots,Z_{[j-1]}$. Equivalently, $Z_{[i]}^* = Z_{[j]}$ will occur if and only if $S_j^* \ge i$ and $S_{j-1}^* < i$. Because $S_{j-1}^* \ge i$ implies $S_j^* \ge i$,

$$\begin{aligned} \Pr(Z_{[i]}^* = Z_{[j]}|F_n) &= \Pr(S_j^* \ge i, S_{j-1}^* < i) \\ &= \Pr(S_j^* \ge i) - \Pr(S_{j-1}^* \ge i). \end{aligned} \tag{5}$$

The cost of computing $\Pr(Z_{[i]}^* = Z_{[j]}|F_n)$ can be reduced by evaluating the beta cumulative distribution function instead of computing and summing the $2(n-j+1)$ binomial probabilities required of formula (5). Specifically, $\Pr(S_j^* \ge i) = F_{i,n-i+1}(j/n)$, where $F_{\alpha,\beta}(x)$ is the cumulative distribution function of a beta random variable with parameters α and β evaluated at x (Mood et al. 1974). Hence,

$$\Pr(Z_{[i]}^* = Z_{[j]}|F_n) = F_{i,n-i+1}(j/n) - F_{i,n-i+1}(j/n - 1/n), \tag{6}$$

and the expectation of the ith bootstrap order statistic is

$$\begin{aligned}\mathrm{E}(y^*_{[i]}|F_n) &= \sum_{j=1}^{n} \Pr(Z^*_{[i]} = Z_{[j]}|F_n) y_{[j]} \\ &= \sum_{j=1}^{n} [F_{i,n-i+1}(j/n) - F_{i,n-i+1}(j/n - 1/n)] y_{[j]}. \end{aligned} \tag{7}$$

While the beta cumulative distribution function does not have a general closed-form expression, numerical approximations are accurate and widely available within most statistical and mathematical software.

Hutson and Ernst (2000) present a formula similar to (6) for the bootstrap probability $\Pr(Z^*_{[i]} = Z_{[j]}|F_n)$, though their derivation is quite different from that presented above. In addition, they present formulae for the exact bootstrap expectation and variance of an L-estimator. In fact, the elementary k-nearest neighbor learner $\eta(X|\mathbf{Z}) = w^T Y_{1:n}$ is an L-estimator, though the ordering on the vector $Y_{1:n}$ is not determined by $y_1, \ldots, y_n$ but is instead induced by the distances between the target input X and the training sample inputs $X_1, \ldots, X_n$. Returning to the formulation of the exact bagging k-nearest neighbor learner, (3) and (7) together yield

$$\eta^*(X|\mathbf{Z}) = \sum_{j=1}^{n} y_{[j]} \sum_{i=1}^{n} w_i [F_{i,n-i+1}(j/n) - F_{i,n-i+1}(j/n - 1/n)]. \tag{8}$$

Caprile et al. (2004) and Steele et al. (2003) have derived other formulae for $\mathrm{E}[\eta(X|\mathbf{Z}^*)|F_n]$. The computational demands of these formulae are substantially greater than formula (8).

3.1.1 The exact bagging k-nearest neighbor classifier

Suppose that the multinomial output Y identifies the class membership of Z. Accordingly, Y and $Y_i, i = 1, \ldots, n$ are a multinomial vectors of length c. In this situation, the order statistic induced by the distances between the target input X and the sample inputs is the $n \times c$ matrix $\mathbf{Y}_{1:n}$ set up in (1) and the k-nearest neighbor learner is an L-estimator of $\pi(X|Z)$. The exact bootstrap expectation of $\eta(X|\mathbf{Z})$ is also an L-estimator of $\pi(X|Z)$ and can be expressed as

$$\begin{aligned}\eta^*(X|\mathbf{Z}) &= \underset{1\times n}{w^T} \mathrm{E}(\underset{n\times c}{\mathbf{Y}_{1:n}}|F_n) \\ &= w^T \mathbf{P} \mathbf{Y}_{1:n}, \end{aligned} \tag{9}$$

where $\mathbf{P}$ denotes the $n \times n$ matrix with $\Pr(Z^*_{[i]} = Z_{[j]}|F_n)$ (formula (6)) in the ith row and jth column. Herein, the exact bagging k-nearest neighbor classification rule assigns Z to the class for which the estimated probability of class membership is largest; specifically, the prediction of class membership is $\arg\max_j \eta^*_j(X|\mathbf{Z})$. This procedure coincides with the usual voting scheme used in conventional bagging in which each bootstrap learner produces a prediction and the class most frequently predicted among the B predictions is taken to be the ensemble prediction. To understand why the exact and conventional algorithms coincide, suppose that class g is most likely to be the prediction of a bootstrapped k-nearest neighbor learner across all possible bootstrap learners $\eta(X|\mathbf{Z}^*)$. Then, class g is the most likely prediction of a Monte Carlo bagged learner $\eta(X|\mathbf{Z}^{*B})$. Moreover, class g is the class with the maximum estimated probability of membership over F_n and hence, also the prediction of the exact bagging k-nearest neighbor learner.

3.2 Sub-sampling

Bootstrap sub-aggregation is carried out by sampling $m < n$ observations randomly from F_n (Bickel et al. 1997; Bühlmann and Yu 2002; Hall and Samworth 2005). For the bagged *nearest* neighbor classifier ($k = 1$), Biau et al. (2008) and Hall and Samworth (2005) have presented asymptotic arguments showing that substantial reductions in prediction error are possible under bootstrap sub-sampling. In practice, enlarging the set of candidate learners to encompass bootstrap sub-aggregation substantially increases the computational effort of searching for a best k-nearest neighbor learner, particularly if the bagged learners are constructed via a Monte Carlo algorithm. It is useful then to develop exact analytic formulae for these learners and thereby avoid Monte Carlo simulation. This section develops analytic formulae for computing the exact bootstrap sub-aggregated k-nearest neighbor learner.

Again, for simplicity, suppose that the output variable is scalar and quantitative and that $m < n$ observations are sampled randomly and *with replacement* from F_n. In this situation, the m bootstrap inputs $X_i^*, i = 1, \dots, m$, and the target input X induce the order statistic $\mathbf{Z}_{1:m}^* = (Z_{[1]}^*, \dots, Z_{[m]}^*)$. Hence, the weight vector w is also of length m and the k-nearest neighbor learner is $\eta(X|\mathbf{Z}^*) = w^T Y_{1:m}^*$. The exact bootstrap sub-aggregated k-nearest neighbor learner is

$$\begin{aligned}\eta^*(X|\mathbf{Z}) &= \mathrm{E}(w^T Y_{1:m}^* | F_n)\\ &= \sum_{i=1}^{m} w_i \sum_{j=1}^{n} \Pr(Z_{[i]}^* = Z_{[j]} | F_n) y_{[i]}^*.\end{aligned}$$

The event $Z_{[i]}^* = Z_{[j]}$ differs from full bootstrap sampling only to the extent that there are m binomial trials in which to sample from $\{Z_{[1]}, \dots, Z_{[j]}\}$. Consequently, the derivation $\Pr(Z_{[i]}^* = Z_{[j]} | F_n)$ parallels the derivation under full bootstrap sampling and leads to the formula

$$\Pr(Z_{[i]}^* = Z_{[j]} | F_n) = F_{i,m-i+1}(j/n) - F_{i,m-i+1}(j/n - 1/n). \tag{10}$$

The exact bagging k-nearest learner can be expressed as $\eta^*(X|\mathbf{Z}) = w^T \mathbf{P} Y_{1:n}$ where $\mathbf{P}_{ij} = \Pr(Z_{[i]}^* = Z_{[j]} | F_n), i = 1, \dots, m$ and $j = 1, \dots, n$ are defined by formula (10).

Now suppose that sampling is *without replacement*. If $i > j$ or $i > m$, then $\Pr(Z_{[i]}^* = Z_{[j]}) = 0$. Suppose that $j \geq i$; then $Z_{[i]}^* = Z_{[j]}$ if and only if $i - 1$ observations are drawn from $\{Z_{[1]}, \dots, Z_{[j-1]}\}$ and $m - i$ observations are drawn from $\{Z_{[j+1]}, \dots, Z_{[n]}\}$. Hence, if $j \geq i$ and $i \leq m$,

$$\Pr(Z_{[i]}^* = Z_{[j]} | F_n) = \frac{\binom{j-1}{i-1}\binom{n-j}{m-i}}{\binom{n}{m}}. \tag{11}$$

Now $\eta^*(X|\mathbf{Z}) = w^T \mathbf{P} Y_{1:n}$ where the entries of $\mathbf{P}$ are given by formula (11).

Lastly, suppose that the output variable identifies class membership so that the matrix form of the k-nearest neighbor learner is $\eta(X|\mathbf{Z}) = w^T \mathbf{Y}_{1:n}$ where the $n \times c$ matrix $\mathbf{Y}_{1:n}$ is the order statistic. The exact bootstrap expectation of $\eta(X|\mathbf{Z})$ under sub-sampling, either with or without replacement is again $\eta^*(X|\mathbf{Z}) = w^T \mathbf{P} \mathbf{Y}_{1:n}$ except now $\mathbf{P}$ is defined by either equation (10) or (11), depending on which sampling scheme is adopted.

3.3 k-nearest neighbor weights

Consider the elementary k-nearest neighbor learner with weights $w_i = k^{-1}$ if $i \leq k$ and $w_i = 0$ if $i > k$. The exact bootstrap expectation is

$$\eta^*(X|\mathbf{Z}) = k^{-1} \sum_{j=1}^{n} y_{[j]} \sum_{i=1}^{k} \Pr(Z^*_{[i]} = Z_{[j]}|F_n), \tag{12}$$

though the bootstrap probabilities $\Pr(Z^*_{[i]} = Z_{[j]}|F_n)$ depend on the method of bootstrap sampling. From (12), it is apparent that $\eta^*(X|\mathbf{Z})$ is a weighted mean of $y_{[1]}, \ldots, y_{[n]}$ where the weight associated with the jth nearest neighbor $y_{[j]}$ is $k^{-1} \sum_{i=1}^{k} \Pr(Z^*_{[i]} = Z_{[j]}|F_n)$. This weight is the average probability that $Z_{[j]}$ will be the ith nearest neighbor of Z under bootstrap sampling, for $1 \leq i \leq k$. These weights are of some interest as they reveal how bagging operates and bagged k-nearest neighbor learners differ from conventional k-nearest neighbor learners. Figures 1 and 2 graph the weights as a function of k and sampling scheme. In Fig. 1, the weights are plotted against j under sampling without replacement and for a sample size of $n = 20$ and $k \in \{1, 3, 5, 8\}$. Figure 2 is the same as Fig. 1 except that the sample size is $n = 200$. The corresponding figures under sampling *with* replacement are omitted because the relationships among k, sampling fraction and the weights are quite similar to those shown in Figs.1 and 2. Figures 1 and 2 show that the weight associated with $Z_{[j]}$ depends on j, k and the bootstrap sampling scheme. For all m, k and j, the exact bagging weights are greater than 0 and less than $1/k$ in contrast to the elementary k-nearest neighbor weights which are either 0 or $1/k$. Hence, bagging acts on k-nearest neighbor learners by smoothing, and the practical effect of smoothing is to reduce the influence of $Z_{[j]}$, $j \leq k$ and to increase the influence of $Z_{[j]}$, $j > k$. For fixed k, smaller sub-sampling fractions (m/n) induce greater degrees of smoothing.

The smoothing effect of bootstrap sub-sampling implies that when Monte Carlo bootstrap sampling is employed, small sub-sampling fractions tend to generate learners $\eta(\cdot|\mathbf{Z}^{*b})$ that differ from the elementary learner $\eta(\cdot|\mathbf{Z})$ to a greater extent than bootstrap learners generated without sub-sampling $(m = n)$. It is sometimes argued (for example, Breiman 1996) that a superior ensemble learner is one in which the constituent learners simultaneously are as different as possible and individually accurate. Figures 1 and 2 show that sub-sampling does produce differences among constituent learners, however, it should be noted that the accuracy of the constituent learners may decline substantially if n is small or if more distant observations are of limited value for prediction. Remark: the exact bagging 1-nearest neighbor weights are simply $\Pr(Z^*_{[1]} = Z_{[j]}|F_n)$, and are similar to the neighbor weights used in a kernel density classifier with a Gaussian kernel and data-dependent width (see Hastie et al. 2001, Chap. 6). This correspondence suggests similarities between the performance of kernel density classifiers and exact bagging 1-nearest neighbors.

3.3.1 Asymptotic bootstrap probabilities of the 1*-nearest neighbor learner*

Asymptotic formulae for the bootstrap probabilities associated with the exact 1-nearest neighbor learner are investigated in this section. These formulae provide further insight into mechanism of bootstrap sub-sampling; moreover, asymptotic versions of the exact bagging 1-nearest neighbor learner are potentially useful base learners in the construction of ensemble learners because of their simplicity and accuracy. Gertheiss and Tutz (2008) and Pančov and Džeroski (2007) provide examples of constructing ensembles using nearest neighbors learners.

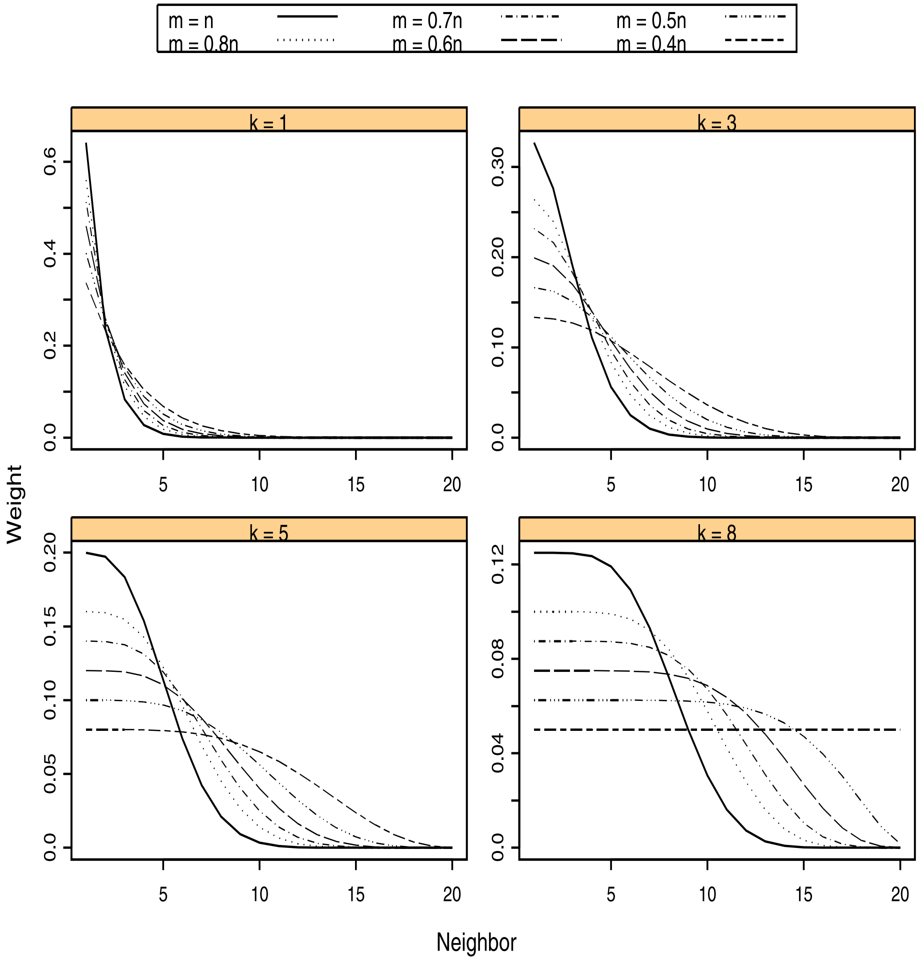

Fig. 1 Exact bootstrap weights for each neighbor given $k \in \{1, 3, 5, 8\}$, sub-sample size m, and a training sample size of $n = 20$. Note that the vertical scale differs among panels

First consider prediction of a quantitative output using the exact bagging 1-nearest neighbor learner

$$\eta^*(X|\mathbf{Z}) = \sum_{j=1}^{n} y_{[j]} \Pr(Z^*_{[1]} = Z_{[j]}|F_n). \tag{13}$$

Now consider conventional bootstrap sampling with replacement. For this case, the bootstrap probabilities are

$$\begin{aligned}\Pr(Z^*_{[1]} = Z_{[j]}|F_n) &= \Pr(S^*_{j-1} = 0) - \Pr(S^*_j = 0)\\ &= \left(\frac{n-j+1}{n}\right)^n - \left(\frac{n-j}{n}\right)^n,\end{aligned} \tag{14}$$

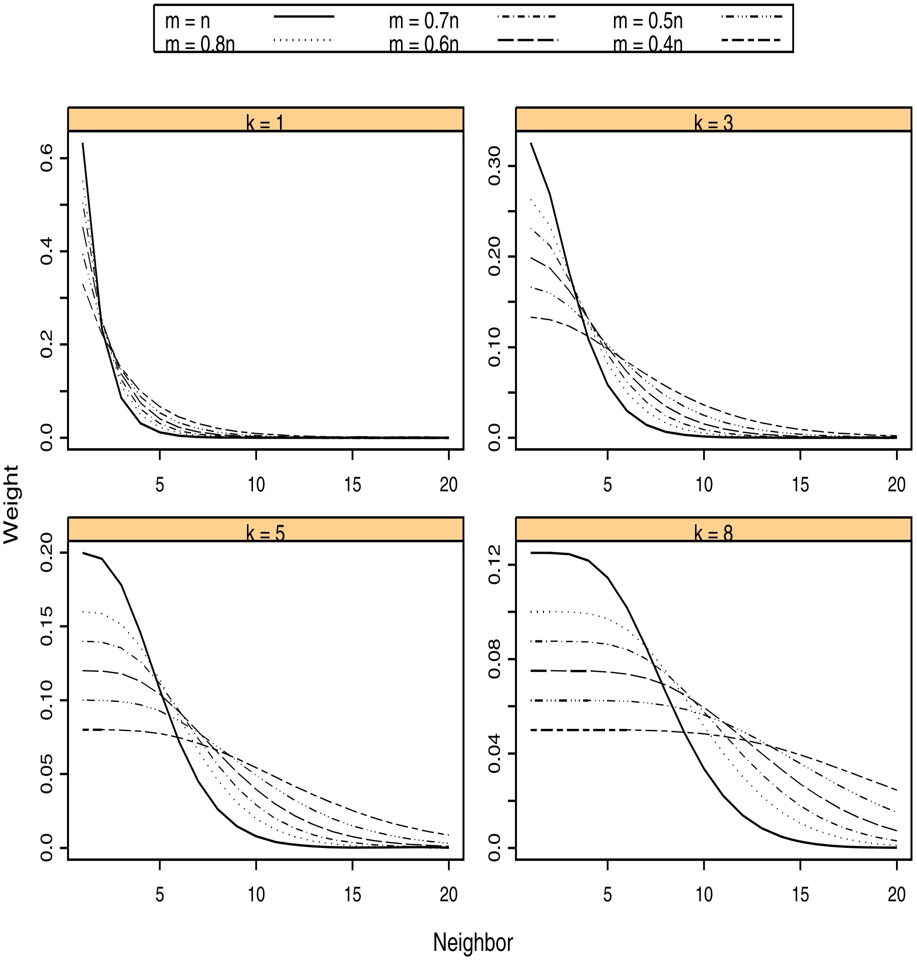

Fig. 2 Exact bootstrap weights for the nearest 20 neighbors for $k \in \{1, 3, 5, 8\}$, sub-sample size m, and a training sample size of $n = 200$. Note that the vertical scale differs among panels

for $j = 1, \dots, n$. An asymptotic approximation to the exact bagging 1-nearest neighbor can be found by taking the limit of the right-hand side of (14) allowing $n \to \infty$. The exact bootstrap expectation of the 1-nearest neighbor learner and its asymptotic equivalent are

$$\eta^*(X|\mathbf{Z}) = \sum_{j=1}^{n} \left[\left(\frac{n-j+1}{n} \right)^n - \left(\frac{n-j}{n} \right)^n \right] y_{[j]}$$

$$\approx \sum_{j=1}^{n} (e^{-j+1} - e^{-j}) y_{[j]}.$$

Now consider bootstrap sub-sampling with replacement where m observations are sampled from $\mathbf{Z}$ and $\alpha = m/n$ denotes the sub-sampling fraction. The bootstrap probabilities again can be expressed in terms of $\Pr(S_j^* = 0)$ and $\Pr(S_{j-1}^* = 0)$ where $S_j^* \sim \mathrm{Bin}(m, j/n)$

and $S^*_{j-1} \sim \mathrm{Bin}(m, (j-1)/n)$. Then,

$$\Pr(Z^*_{[1]} = Z_{[j]}|F_n) = \left[\left(\frac{n-j+1}{n}\right)^{\alpha n} - \left(\frac{n-j}{n}\right)^{\alpha n}\right], \quad j = 1, \dots, m,$$

and the asymptotic approximation of the exact bagging 1-nearest neighbor learner under sub-sampling with replacement is $\eta^*(X|\mathbf{Z}) \approx \sum_{j=1}^n (e^{-\alpha(j-1)} - e^{-\alpha j})y_{[j]}$.

Lastly, consider bootstrap sub-sampling without replacement where m observations are sampled from $\mathbf{Z}$ and $\alpha = m/n$ denotes the sub-sampling fraction. The bootstrap probabilities are

$$\begin{aligned}\Pr(Z^*_{[1]} = Z_{[j]}|F_n) &= \frac{\binom{n-j}{m-1}}{\binom{n}{m}}\\ &= \frac{n-m}{n}\frac{n-m-1}{n-1} \times \cdots \times \frac{n-m-j+2}{n-j+2}\frac{m}{n-j+1},\end{aligned}$$

$$j = 1, \dots, m.$$

With $m = \alpha n$,

$$\lim_{n\to\infty} \Pr(Z^*_{[1]} = Z_{[j]}|F_n) = \alpha(1-\alpha)^{j-1}, \quad j = 1, \dots, m.$$

The asymptotic approximation of the exact bagging 1-nearest neighbor learner under sub-sampling without replacement is $\eta^*(X|\mathbf{Z}) \approx \sum_{j=1}^n \alpha(1-\alpha)^{j-1} y_{[j]}$.

The effect of sampling fraction α on the asymptotic exact bagging 1-nearest neighbor is revealed by plotting the bootstrap weights $\Pr(Z^*_{[1]} = Z_{[j]}|F_n)$ against α. Figure 3 shows the bootstrap weights for the 6 nearest neighbors under sampling with and without replacement. Differences in bootstrap weights between sampling scheme are small except when the sampling fraction is greater than 0.5, and then the largest differences occur with the first and second nearest neighbors. Consider the bootstrap weights for the nearest neighbor $Z_{[1]}$. As the sampling fraction decreases from 1 towards 0, the weights of decrease monotonically under both sampling schemes. In contrast, the bootstrap weights for neighbors $Z_{[2]}, \dots, Z_{[6]}$ increase and then decay towards zero as α approaches zero. This decay and convergence towards zero for all bootstrap weights as $\alpha \to 0$ occurs because $\lim_{\alpha\to 0} \Pr(Z^*_{[1]} = Z_{[j]}|F_n) = 0$ and $\lim_{\alpha\to 0} \Pr(Z^*_{[1]} = Z_{[j]}|F_n)/\Pr(Z^*_{[1]} = Z_{[j-1]}|F_n) = 1$ for $j \geq 2$ under both sampling schemes.

4 *k*-nearest neighbor regression learners

Suppose that the output variable y is quantitative and its expectation is a linear function of the concomitant input vector X. A local regression approach via k-nearest neighbor regression (Altman 1992; Cleveland and Devlin 1988; Loader 1999) may be useful if the linear function varies over the input space instead of being globally constant. A varying linear function is accommodated within the k-nearest neighbor framework as follows. Let Z_0 denote the target and suppose that $\mathrm{E}(y_0|X_0) = X_0^T\beta_0$ where β_0 is an unknown vector of coefficients. Suppose further that the locally linear model $\mathrm{E}(y_{[j]}|X_{[j]}) = X_{[j]}^T\beta_0$ holds for $j = 1, \dots, k$ where $\{Z_{[1]}, \dots, Z_{[k]}\}$ are the k-nearest neighbors of Z_0. In principle, a linear model may be fit using the k nearest neighbors and a prediction computed using the fitted model which improves on the conventional k-nearest neighbor prediction. Though the locally linear model

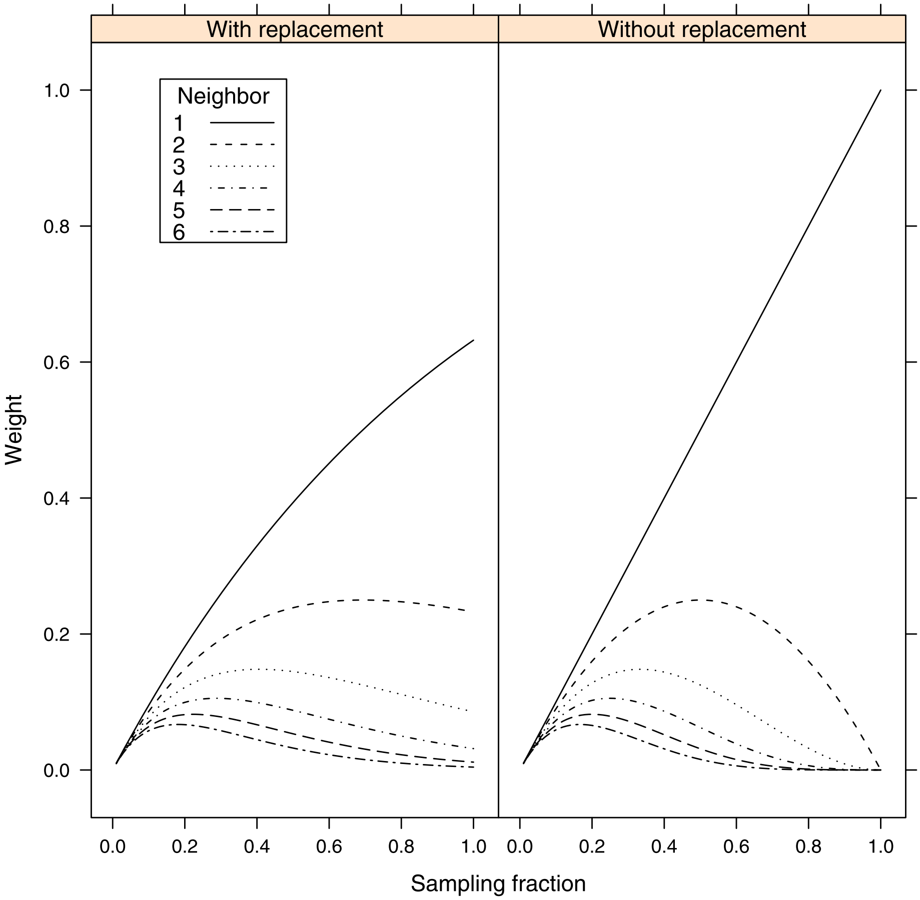

Fig. 3 Asymptotic weights $\Pr(Z^*_{[1]} = Z_{[j]}|F_n)$ as a function of the sampling fraction for neighbors $j = 1, \dots, 6$ for the asymptotic exact bootstrap 1-nearest neighbor learner when sampling with and without replacement

is somewhat contrived, there are situations in which the model is approximately correct for prediction of the intended target inputs. For example, the locally linear model is correct if a global linear model is correct though the k-nearest neighbor regression learner is not optimal by the least squares criterion. Another example is the situation in which the relationship between one or more input variables and the expected response variable is continuous, though nonlinear. Then, restricting the inputs to the nearest k-neighbors may yield a linear approximation of the true model that is reasonably accurate in a local neighborhood of the input X.

To proceed, let $\psi_0(Z)$ denote the indicator function of the event $Z \in \{Z_{[1]}, \dots, Z_{[k]}\}$ and $\boldsymbol{\Psi}$ denote a diagonal matrix with diagonal $[\psi_0(Z_1), \dots, \psi_0(Z_n)]$. Also, let $\mathbf{X} = (X_{ij})$ denote the $n \times p$ matrix constructed from $X_1, \dots, X_n$, and let $Y = (y_1, \dots, y_n)^T$ denote the vector of training sample outputs. The least squares estimator of β_0 is obtained by minimizing the objective function

$$S(\beta_0|\mathbf{Z}) = (Y - \mathbf{X}\beta_0)^T \boldsymbol{\Psi}(Y - \mathbf{X}\beta_0) \tag{15}$$

with respect to β_0. Provided that $\mathbf{X}^T\Psi\mathbf{X}$ is nonsingular and the locally linear model $\mathrm{E}(y_{[j]}|X_{[j]}) = X_{[j]}^T\beta_0$ $j = 1, \dots, k$, is correct, then the least squares estimator of β_0 is $\widehat{\beta}_0 = (\mathbf{X}^T\boldsymbol{\Psi}\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\Psi}Y$. The k-nearest neighbor regression learner predicts y_0 by $\eta(X_0|\mathbf{Z}) = X_0^T\widehat{\beta}_0$. When k is small, the variance of $\widehat{\beta}_0$ may be large and the learner unstable; worse, when $k \approx p$, $\mathbf{X}^T\boldsymbol{\Psi}\mathbf{X}$ often will be ill-conditioned or singular. Moreover, the training sample inputs $X_{[1]}, \dots, X_{[k]}$ tend to be close to the mean vector $k^{-1}\sum_{i=1}^{k} X_{[i]}$ by virtue of being close to X, and this contributes to the instability of the learner. For more than a few of the comparison data sets discussed below, ill-conditioning was a problem for $k \le 20$. Two different modifications of the k-nearest neighbor regression learner aimed at alleviating ill-conditioning and reducing instability follow.

Ridge regression, or more generally, regularization, is an effective method for reducing instability and accommodating less-than-full rank design matrices (Friedman 1989; Hastie et al. 2001; Hoerl and Kennard 1970; Loh 1995). The ridge regression estimator replaces $\mathbf{X}^T\mathbf{X}$ with $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ where $\lambda > 0$ is chosen to insure that the determinant of $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ is non-zero. Relative to the least squares estimator, the ridge regression tends to shrink the Euclidean norm of $\widetilde{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T Y$ towards 0, and the effect on the learner is to reduce instability. A regularized local estimator $\widetilde{\beta}_0 = (\mathbf{X}^T\boldsymbol{\Psi}\mathbf{X} + 10^{-5}\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{\Psi}Y$ was used in the study discussed below.

There is some hope in substantively improving k-nearest neighbor regression learners by exact bagging because of their instability problems. However, note that the k-nearest neighbor regression learner can be written as $\eta(X|\mathbf{Z}) = aY_{1:n}$ with $a_i = X^T(\mathbf{X}^T\boldsymbol{\Psi}\mathbf{X})^{-1}\mathbf{X}^T\Psi_i$ where Ψ_i is the ith column of $\boldsymbol{\Psi}$. Though the k-nearest neighbor regression learner is a linear combination of $Y_{1:n}$, this learner does not satisfy the conditions necessary for $\mathrm{E}(aY_{1:n}^*|F_n) = a\mathrm{E}(Y_{1:n}^*|F_i)$ because a is not a fixed vector but instead a function of $Z_1, \dots, Z_n$. Consequently, the exact bootstrap expectation of the k-nearest neighbor regression learner does not have a simple closed form. Therefore, an alternate approach is pursued in which the estimator of β_0 is chosen to minimize the exact bootstrap expectation of the objective function $\mathrm{E}[S(\beta_0|\mathbf{Z}^*)|F_n]$. The resulting learner is referred to as a *partially* exact bootstrap k-nearest neighbor regression learner. Theorem 1 identifies the estimator of β_0.

Theorem 1 *If* $\mathbf{X}$ *is full rank, then* $\beta_0^* = E(\mathbf{X}^{*T}\Psi_0^*\mathbf{X}^*|F_n)^{-1}E(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n)$ *minimizes* $E[S(\beta_0|\mathbf{Z}^*)|F_n] = E\{(Y^* - \mathbf{X}^*\beta_0)^T\boldsymbol{\Psi}_0^*(Y^* - \mathbf{X}^*\beta_0)|F_n\}$.

Proof The order of differentiation and integration can be reversed when differentiating $\mathrm{E}[S(\beta_0|\mathbf{Z}^*)|F_n]$ with respect to β_0 because $S(\beta_0|\mathbf{Z}^*)$ [see (15)] is a continuous function of β_0 and F_n is a discrete distribution function with at most n^n points with non-zero probability. Hence

$$\frac{\partial \mathrm{E}[S(\beta_0|\mathbf{Z}^*)|F_n]}{\partial\beta_0} = -2\mathrm{E}\left[\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*(Y^* - \mathbf{X}^*\beta_0)|F_n\right]. \tag{16}$$

Setting the vector of partial derivatives equal to 0 yields the normal equations $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*| F_n) = \mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n)\beta_0$. Suppose now that the $n \times p$ design matrix $\mathbf{X}$ is full rank. Theorem 2 shows that $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*|F_n)$ is positive definite and consequently the unique solution to the normal equations is

$$\beta_0^* = \mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*|F_n)^{-1}\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n). \tag{17}$$

Furthermore, since $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*|F_n)$ is positive definite,

$$\frac{\partial^2 \mathrm{E}[S(\beta_0|\mathbf{Z}^*)|F_n]}{\partial\beta\partial\beta^T} = -\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*|F_n) \tag{18}$$

is negative definite and it follows that β_0^* minimizes $\mathrm{E}[S(\beta_0|\mathbf{Z}^*)|F_n]$. □

Theorem 2 establishes that $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*|F_n)^{-1}$ and $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n)$ are computationally simple.

Theorem 2 *Without loss of generality, assume that the rows of* $\mathbf{X}$, Y, *and* $\boldsymbol{\Psi}$ *have been arranged in ascending order according to the distances between* X_0 *and* $X_1, \ldots, X_n$. *Let* $\mathbf{A}$ *denote a diagonal matrix such that the* j*th diagonal element is* $\sum_{i=1}^k \Pr(Z_{[i]}^* = Z_{[j]}|F_n)$. *Then,* $E(\mathbf{X}^{*T}\Psi^*Y^*|F_n) = \mathbf{X}^T\mathbf{A}Y$ *and* $E(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*|F_n) = \mathbf{X}^T\mathbf{A}\mathbf{X}$. *Furthermore,* $\mathbf{X}^T\mathbf{A}\mathbf{X}$ *is full rank and* $\beta_0^* = (\mathbf{X}^T\mathbf{A}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{A}Y$.

Proof Let $X_j = (x_{1,j}, \ldots, x_{n,j})^T$ denote the jth column of $\mathbf{X}$. Note that the rth diagonal element of $\boldsymbol{\Psi}$ indicates the event $\{r \le k\}$. Then, the i, jth element of $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*|F_n)$ is

$$\begin{aligned}
\mathrm{E}(\mathbf{X}_i^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}_j^*|F_n) &= \sum_{r=1}^k \mathrm{E}(x_{[r],i}^* x_{[r],j}^*|F_n) \\
&= \sum_{r=1}^k \sum_{s=1}^n x_{[s],i} x_{[s],j} \Pr(Z_{[r]}^* = Z_{[s]}|F_n) \\
&= \sum_{s=1}^n x_{[s],i} x_{[s],j} \sum_{r=1}^k \Pr(Z_{[r]}^* = Z_{[s]}|F_n) \\
&= X_i^T \mathbf{A} X_j,
\end{aligned}$$

because $\mathbf{A}$ is diagonal and the sth diagonal element of $\mathbf{A}$ is defined to be $\sum_{r=1}^k \Pr(Z_{[r]}^* = Z_{[s]}|F_n)$. Hence, $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*\mathbf{X}^*|F_n) = \mathbf{X}^T\mathbf{A}\mathbf{X}$.

The calculation of $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n)^T = [\mathrm{E}(X_1^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n), \ldots, \mathrm{E}(X_p^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n)]$ proceeds in the same fashion. The jth element is

$$\begin{aligned}
\mathrm{E}(X_j^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n) &= \sum_{r=1}^k \mathrm{E}(x_{[r],j}^* y_{[r]}^*|F_n) \\
&= \sum_{s=1}^n x_{[s],j} y_{[s]} \sum_{r=1}^k \Pr(Z_{[r]}^* \quad = Z_{[s]}|F_n) \\
&= X_j^T \mathbf{A} Y.
\end{aligned}$$

Thus, $\mathrm{E}(\mathbf{X}^{*T}\boldsymbol{\Psi}_0^*Y^*|F_n) = \mathbf{X}^T\mathbf{A}Y$.

To determine the rank of $\mathbf{X}^T\mathbf{A}\mathbf{X}$ under the assumption that $\mathbf{X}$ is full rank, note that $\mathbf{A}$ is full rank because the diagonal elements of $\mathbf{A}$ are all positive. Hence, $\mathrm{rank}(\mathbf{X}^T\mathbf{A}\mathbf{X}) = \mathrm{rank}(\mathbf{X}^T\mathbf{X})$. Thus, $\mathbf{X}^T\mathbf{A}\mathbf{X}$ is full rank and nonsingular. □

In principle, replacing $\mathbf{X}^T \mathbf{\Psi} \mathbf{X}$ by $\mathrm{E}(\mathbf{X}^{*T} \mathbf{\Psi}_0^* \mathbf{X}^* | F_n) = \mathbf{X}^T \mathbf{A} \mathbf{X}$ will tend to reduce the variance of the estimator of β_0 and improve stability. However, in applications $\mathbf{X}^T \mathbf{A} \mathbf{X}$ is sometimes ill-conditioned when k of the same order as p; for this reason, in the examples discussed below a regularized version of β_0^* given by $(\mathbf{X}^T \mathbf{A} \mathbf{X} + 10^{-5} \mathbf{I})^{-1} \mathbf{X}^T \mathbf{A} Y$ was used.

5 Comparisons of statistical learners

The performance of k-nearest neighbor and related learners are compared using data sets available from the UCI data repository and previously used by Breiman (2001) in comparisons of random forests, Adaboost, and adaptive bagging. Tables 1 and 2 provide a brief summary of the data sets and Breiman (2001) and provides specifics regarding the learners. In the case of a quantitative output variable, cross-validation estimates of mean squared error were obtained by comparing predictions $\eta(X_i | \mathbf{Z}_{-i})$ to y_i, where $\mathbf{Z}_{-i}$ is a training set not containing y_i. For categorical output variables, prediction error was estimated by the percentage of test targets incorrectly predicted by $\eta(X_i | \mathbf{Z}_{-i})$. For most examples, 10-fold cross-validation was carried out by randomly partitioning a set of n observations

Table 1 Summaries and cross-validation details for data sets involving categorical output variables

Set	Number of observations (n)	Training set size	Repetitions	Number of inputs	Number of classes (c)
Breast	699	$[0.9n]$	100	9	2
Diabetes	768	$[0.9n]$	100	8	2
Ecoli	336	$[0.9n]$	100	7	8
Glass	214	$[0.9n]$	100	9	6
Image	2310	$[0.9n]$	100	19	7
Ionosphere	315	$[0.9n]$	100	34	2
Satellite image	6435	4435	1	36	6
Sonar	208	$[0.9n]$	100	60	2
Vehicle	846	$[0.9n]$	100	18	4
Vowel	990	$[0.9n]$	100	10	11

Table 2 Summaries and cross-validation details for data sets involving quantitative output variables

Set	Number of observations (n)	Training set size	Repetitions	Number of inputs (p)
Abalone	4177	$[0.75n]$	10	8
Boston housing	506	$[0.9n]$	100	12
Friedman #1	2200	200	1	10
Friedman #2	2200	200	1	4
Friedman #3	2200	200	1	4
Ozone	330	$[0.9n]$	100	8
Servo	167	$[0.9n]$	100	4

as a training set of $[0.9n]$ observations and a test set of $n - [0.9n]$ observations. Cross-validation was repeated (usually 100 times) and the estimates averaged. Following Breiman (2001), 10 repetitions of 4-fold cross-validation were used with the Abalone data set. The satellite image data set had been previously partitioned as a training set of 4435 observations and a test set of 2000 observations and this scheme was adopted for my comparisons. Error estimates for the synthetic data sets Friedman #1, #2 and #3 were obtained by generating independent training sets of 200 observations and test sets of 2000 observations. Generally, the within-sample standard errors of the estimates are negligible by virtue of averaging multiple cross-validation estimates. In other words, further repetitions of the algorithm will produce no meaningful changes in the error estimates. This does not imply that the addition of new observations to the data sets, or simply a set of new test observations will yield the same estimates; instead, the error estimates presented below are conditional on the sample and formal statistical inferences drawn from these comparisons are necessarily limited to the samples themselves. Efron and Tibshirani (1997) provide a detailed discussion of the problem of estimating the variance of error estimators.

The k-nearest neighbor learners were constructed for $k \in \{1, 5, 10, 20, 50\}$. Additional values of $k = 70$ and 100 were used with Abalone, Boston housing and Ozone data sets after observing that small values of estimated prediction error were associated with larger values of k. The exact bagging k-nearest neighbor and partially exact bagging k-nearest neighbor regression learners were constructed under sub-sampling with and without replacement using sampling fractions $\alpha \in \{0.75, 0.5, 0.25\}$.

Tables 3 and 4 show the prediction error estimates for the categorical and quantitative output variable data sets respectively. Estimates for the elementary and exact bagging k-nearest neighbor learners and the regularized k-nearest neighbor regression and partially exact k-nearest neighbor regression learner are reported as the smallest value over k; estimates for the exact bagging k-nearest neighbor and partially exact k-nearest neighbor regression learners using sub-sampling are also reported as the smallest estimate over k and α. Despite the computational simplicity of the k-nearest neighbor learners, the error estimates are not substantially worse than the comparison learners except for the Ionosphere and Servo data sets, and in several instances (Ozone and Vowel), the k-nearest neighbor learners produced

Table 3 Cross-validation estimates of prediction error. For each of the k-nearest neighbor methods, estimates were computed for $k \in \{1, 5, 10, 20, 50\}$ and the minimum estimate among these 5 estimates is presented below. Estimates for the Adaboost and Random Forest learners were extracted from Breiman (2001)

Set	k-NN	Exact	Sub-sampling		Adaboost	Random
		bagging	with	without		Forest
Breast	2.94	2.94	3.04	2.99	3.2	3.1
Diabetes	23.2	23.5	23.2	23.4	26.6	23.0
Ecoli	13.0	12.9	12.9	13.7	14.8	12.9
Glass	21.9	21.9	21.6	21.6	22.0	24.4
Image	3.46	3.46	3.49	3.49	1.6	1.6
Ionosphere	13.0	13.0	13.5	13.5	6.5	5.5
Satellite image	9.14	9.14	9.21	9.28	8.8	9.1
Sonar	13.0	13.0	13.0	13.0	15.6	13.6
Vehicle	27.9	27.8	27.4	27.5	23.2	23.1
Vowel	1.04	1.04	1.21	1.21	4.1	3.3

Table 4 Cross-validation estimates of prediction error. For each of the k-nearest neighbor methods, estimates were computed for $k \in \{1, 5, 10, 20, 50\}$. The minimum prediction error estimates among all values of k are presented below. Estimates for the adaptive bagging and Random Forest learners were extracted from Breiman (2001)

Set	k-NN	Exact	Reg. k-NN	PEB k-NN regression			Adaptive	Random
		bagging	regression	none	with	without	bagging	Forest
Abalone	4.99	4.95	5.20	5.05	4.95	4.45	4.9	4.6
Boston housing	20.9	16.9	12.8	16.0	12.7	12.8	9.7	10.2
Ozone	10.2	10.2	9.11	9.12	10.0	8.89	17.8	16.3
Servo	0.591	0.572	0.491	0.414	0.541	0.413	0.251	0.246
Friedman #1	9.07	8.68	6.24	6.16	8.58	6.13	4.1	5.7
Friedman #2 × 10^3	33.3	32.6	20.4	20.4	33.5	20.1	21.5	19.6
Friedman #3 × 10^{-3}	39.4	38.2	32.9	31.8	30.8	30.5	24.8	21.6

consistently smaller error estimates than the competitors. Table 3 also shows that exact bagging k-nearest neighbor learners, including those that utilize sub-sampling appear not to be distinguishable from the elementary k-nearest neighbor learner on the basis of prediction error. Table 4 shows that the error estimates for k-nearest neighbor learners (elementary and exact bagging) were always greater than those obtained from the regularized k-nearest neighbor regression learners. Bagging (exact and partially exact) was largely ineffectual as the error estimates do not consistently favor bagging. Similarly, the effectiveness of sub-sampling varied without consistency among data sets. The dearth of experimental evidence of an accuracy advantage to bagging is attributable to several factors, in particular, the adaptive choice of k by cross-validation. Later in this article, comparisons are made between learners with $k = 1$ fixed which show large differences in estimated accuracy and favoring exact bagging versions of the 1-nearest learners. Additionally, the 17 data sets were selected because they had been used by Breiman (2001) to illustrate the performance of random forest learners and do not necessarily reflect the performance of these learners in other situations.

Another look at exact bagging compares error estimates produced by the elementary and exact bagging versions of the k-nearest neighbor learners across a range of values for k. Figures 4 and 5 show that cross-validation estimates for the exact bagging and partially exact bagging learners are usually less than or nearly equal to corresponding learner constructed without bagging. These data indicate that for a fixed choice of k, exact bagging tends to improve on the elementary k-nearest neighbor learner. However, when looking over a range of values for k which encompass values that yield near-optimal values of cross-validation estimates of error, then the differences in prediction error largely disappear.

5.1 Performance of the asymptotic exact bagging 1-nearest neighbor learner

The asymptotic exact bagging 1-nearest neighbor learner is compared to several elementary k-nearest neighbor learners in this section. In this comparison, no effort is directed towards selecting apparent optimal values of k or sub-sampling fractions α. Instead, three values of k (1, 5 and 20) were used as are three sub-sampling fractions ($\alpha \in \{0.25, 0.5, 1\}$). For quantitative output variables, Table 5 shows that the asymptotic approximation of the exact bagging 1-nearest neighbor under full sampling with replacement ($\alpha = 1$) yields consistently

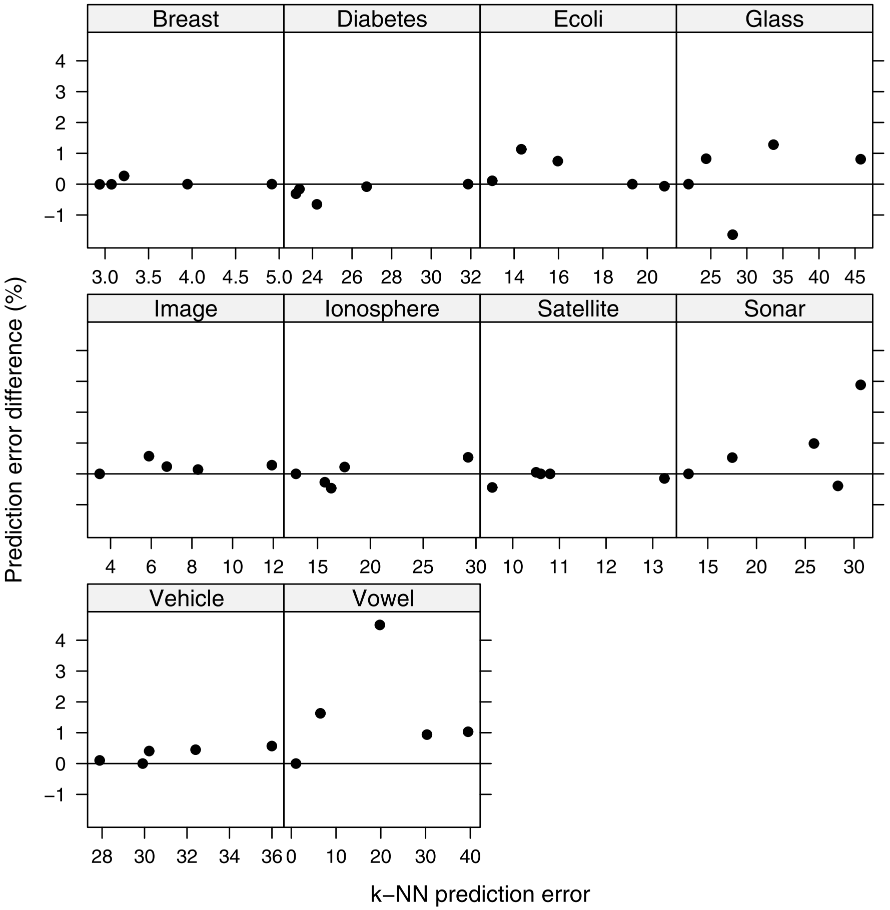

Fig. 4 Prediction error produced by the exact bagging and elementary k-nearest neighbor learners plotted against k

and substantial reductions in estimated prediction error in comparison to the elementary 1-nearest neighbor. In addition, the asymptotic approximation of the exact bagging 1-nearest neighbor under sub-sampling with or without replacement produced additional reductions in estimated error. For example, the estimated prediction error of the sub-sampling versions of the asymptotic exact bagging learners was less than 60% of the estimated error of the elementary learner for the Ozone, Servo, and Friedman #1 data sets and less than 66% for Friedman #2 and #3. Comparing sampling strategies reveals less consistent differences in estimated prediction error between $\alpha = 0.5$ and $\alpha = 0.25$, though, on balance, $\alpha = 0.25$ yielded smaller estimated error. In summary, the examples suggest that the asymptotic exact bagging 1-nearest neighbor learner has the potential for replacing the elementary 1-nearest neighbor learner in applications in which tuning k is not feasible such as constructing ensemble learners from a large collection of simple base learners. These results suggest ensemble strategy in which some or all of the base learners are asymptotic exact bagging 1-nearest neighbor learners using randomly sampled values of α.

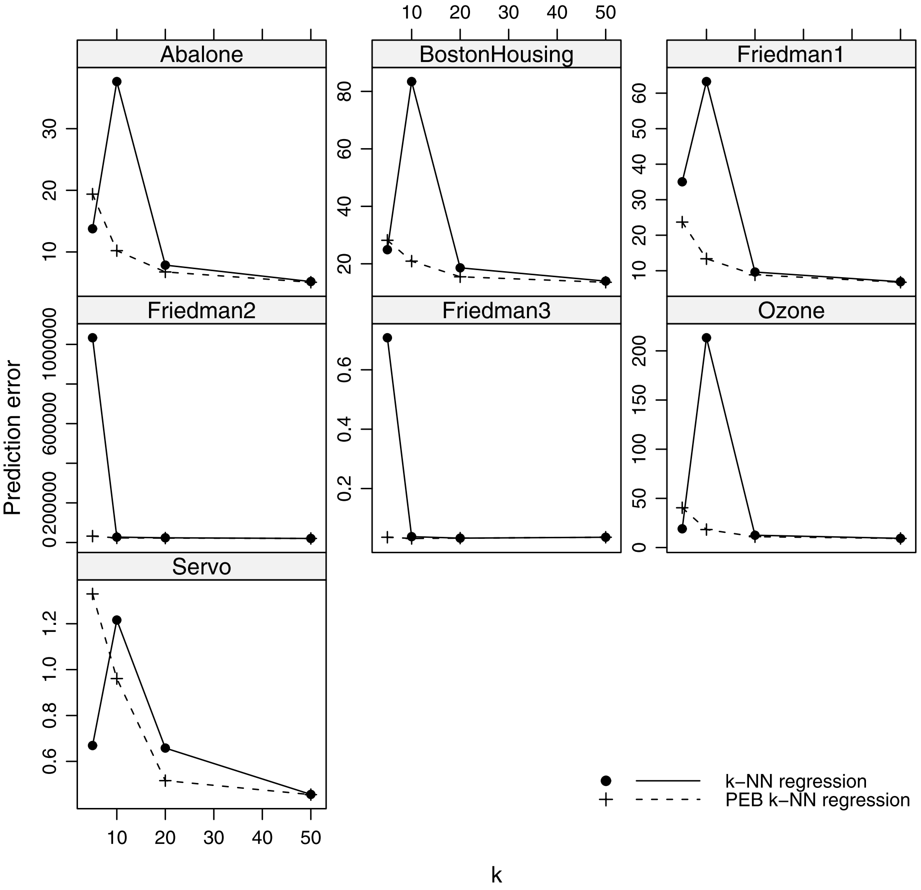

Fig. 5 Prediction error produced by the partially exact bagging and regularized k-nearest neighbor regression learners plotted against k

Elementary k-nearest neighbor learners and asymptotic exact bagging 1-nearest neighbor learners under sampling with and without replacement are compared for prediction of categorical outputs in Table 6. The comparison differs slightly with the comparison of quantitative outputs shown in Table 5, as the sampling fractions were selected to be $\alpha = 1/2$ and $1/4$ when sampling with replacement and $1/3$ and $1/6$ when sampling without replacement. Larger values of α produce weights for the nearest neighbor that lead to the same learner as the elementary 1-nearest neighbor because the output values are limited to 0 or 1. For example, when sampling without replacement with $\alpha = 0.5$, the weight assigned to the nearest neighbor $Y_{[1]}$ is 0.5 and when sampling with replacement and using $\alpha = 1$, the weight is 0.632; hence, the predictions are always the same as the class membership of $Z_{[1]}$. Table 6 shows that for these 10 classification problems, sub-sampling tends to yield smaller estimates of prediction error provided that the best choice of k for the elementary k-nearest neighbor is greater than 1. When the smallest cross-validation estimates of error for the elementary k-nearest neighbor occur with $k = 1$, then all other learners produce larger estimates of prediction error. This result is consistent with Biau et al. (2008) who show that

Table 5 Cross-validation estimates of prediction error for the elementary k-nearest neighbor learner and the asymptotic exact bagging 1-nearest neighbor. Sampling fractions for the asymptotic exact bagging learner are denoted by α

Set	Elementary k-NN			Asymptotic exact bagging 1-NN				
				With replacement		Without rep.		
	$k=1$	$k=5$	$k=20$	$\alpha=1$	$\alpha=1/2$	$\alpha=1/4$	$\alpha=1/3$	$\alpha=1/6$
Abalone	8.62	5.29	5.00	6.38	5.48	5.05	5.33	4.97
Boston housing	23.0	23.0	26.1	19.2	19.3	20.7	17.9	20.6
Ozone	18.2	11.0	10.4	12.6	10.8	10.1	10.6	10.1
Servo	0.938	0.635	1.06	0.767	0.654	0.711	0.548	0.707
Friedman #1	17.8	9.67	11.3	10.8	9.06	8.92	9.17	9.38
Friedman #2 × 10^3	65.7	35.5	45.3	44.3	36.9	35.2	34.9	35.6
Friedman #3 × 10^{-3}	63.9	36.5	45.6	43.2	36.3	34.9	35.4	36.1

Table 6 Cross-validation estimates of prediction error for the elementary k-nearest neighbor learner and the asymptotic exact bagging 1-nearest neighbor. Sampling fractions for the asymptotic exact bagging learner are denoted by α

Set	Elementary k-NN			Asymptotic exact bagging 1-NN			
				With replacement		Without rep.	
	$k=1$	$k=5$	$k=20$	$\alpha=1/2$	$\alpha=1/4$	$\alpha=1/3$	$\alpha=1/6$
Breast	4.46	3.25	3.106	3.62	3.22	3.38	3.18
Diabetes	31.4	26.0	24.3	27.5	25.2	27.1	24.9
Ecoli	19.3	14.5	15.1	15.2	13.4	14.2	13.2
Glass	21.9	23.9	33.9	23.1	24.5	22.9	27.0
Image	3.48	5.69	8.14	4.00	4.95	4.31	5.48
Ionosphere	13.6	15.9	16.1	14.9	16.5	15.1	17.1
Satellite image	27.7	27.9	27.6	27.9	28.0	27.9	28.0
Sonar	13.3	18.7	28.5	14.6	16.8	15.0	19.1
Vehicle	13.1	18.7	28.4	14.6	16.9	14.4	19.4
Vowel	1.27	6.62	13.0	2.42	3.82	2.79	5.46

error rate of the bagged 1-nearest neighbor converges to the error rate of the Bayes classifier provided the sub-sampling fraction is sufficiently small.

6 Discussion

A complete understanding of why bagging works has been elusive. Studies of bagging tend to be either asymptotic in nature, concentrating on bias and variance (e.g., Bühlmann and Yu 2002; Friedman and Hall 2007), and prediction error (Hall and Samworth 2005), or empirical comparisons of bagging performance (Bauer and Kohavi 1999; Maclin and Opitz 1997). A recurrent theme of these studies is that prediction error can be decomposed into variance and bias components, and bagging success is largely attributable to variance reduction. The effect of bagging on bias is uncertain, as a number of contradictory findings have been reported. It has also been argued that bagging success is attributable, at least in part, to smoothing (Bühlmann and Yu 2002) or equalization of training observation influence (Grandvalet 2004).

The elementary k-nearest neighbor is distinguished by the minimal extent to which the learner varies with small perturbations in the data set, a property referred to as stability (Bühlmann and Yu 2002; Buja and Stuetzle 2006). Generally, the k-nearest neighbor learner is stable because a training observation Z_i affects a prediction only when Z_i is one of the k nearest neighbors of the target observations. Usually k is much smaller than the number of training observations so that the influence of Z_i is limited to a local neighborhood about Z_i. When k is relatively large, then each of the k neighbors has an equal (and small) contribution towards a prediction. Operationally, bagging forces the elementary weights $w_i \in \{0, k^{-1}\}$ defining the learner $\eta(X|\mathbf{Z}) = w^T Y_{[1:n]}$ towards n^{-1}. The degree of change is necessarily small when k is large, and when k is small, relatively few weights are substantively changed. Consequently, the predictions of the bagged k-nearest neighbor learner tend to be similar to those of its conventional counterpart.

Acknowledgements I thank the reviewers for comments and suggestions that greatly improved the article and Dave Patterson for many insightful discussions on the subjects of exact bagging and statistical learners.

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *American Statistician*, *46*, 175–185.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Journal of Machine Learning*, *36*, 105–139.

Bhattacharya, P. K. (1974). Convergence of sample paths of normalized sums of induced order statistics. *Annals of Statistics*, *2*, 1034–1039.

Biau, G., Devroye, L., & Lugosi, L. (2008). *Consistency of random forests and other averaging classifiers*. Preprint, University Paris VI.

Bickel, P. J., Götze, F., & van Zwet, W. R. (1997). Resampling fewer than n observations: gains, losses, and remedies for losses. *Statistica Sinica*, *7*, 1–31.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Bühlmann, P., & Yu, B. (2002). Analyzing bagging. *Annals of Statistics*, *30*, 927–961.

Buja, A., & Stuetzle, W. (2006). Observations on bagging. *Statistica Sinica*, *16*(2), 323–352.

Caprile, B., Merler, S., Furlanello, C., & Jurman, G. (2004). Exact bagging with k-nearest neighbor classifiers. In F. Roli, J. Kittler, & T. Windeatt (Eds.), *LNCS: Vol. 3077. MCS 2004* (pp. 72–81). Berlin: Springer.

Cleveland, W. S., & Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, *83*, 596–610.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*, 21–27.

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, *92*, 548–560.

Friedman, J. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, *84*, 165–175.

Friedman, J., & Hall, P. (2007). On bagging and nonlinear estimation. *Journal of Statistical Planning and Inference*, *137*(3), 669–683.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, *38*(2), 337–374.

Gertheiss, J., & Tutz, G. (2008). *Feature selection and weighting by nearest neighbor ensembles* (Technical Report Number 033). Department of Statistics, University of Munich.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York: Springer.

Hutson, A. D., & Ernst, M. D. (2000). The exact bootstrap mean and variance of an L-estimator. *Journal of the Royal Statistical Society, Series B*, *62*, 89–94.

Freund, Y., & Schapire, R. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and Systems Sciences*, *5*, 119–139.

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

Hall, P., & Samworth, R. L. (2005). Properties of bagged nearest neighbor classifiers. *Journal of the Royal Statistical Society, Series B*, *67*(3), 363–379.

Hoerl, A. E., & Kennard, R. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55–67.

Loader, C. (1999). *Local regression and likelihood*. New York: Springer.

Loh, W. L. (1995). On linear discriminant analysis with adaptive ridge classification rules. *Journal of Multivariate Analysis*, *53*, 264–278.

Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. In *Proceedings of the fourteenth national conference on artificial intelligence* (pp. 546–551), Providence, RI.

Mood, A. M., Graybill, F. A., & Boes, D. C. (1974). *An introduction to the theory of statistics*. New York: Wiley.

Pančov, P., & Džeroski, S. (2007). Combining bagging and random subspaces to create better ensembles. In *Advances in intelligent data analysis VII* (pp. 118–129). Berlin: Springer.

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Skurichina, M., & Duin, R. P. W. (1998). Bagging for linear classifiers. *Pattern Recognition*, *31*, 909–930.

Steele, B. M., Patterson, D. A., & Redmond, R. L. (2003). Toward estimating thematic map accuracy without a probability test sample. *Ecological and Environmental Statistics*, *10*, 333–356.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.