

Exact bootstrap k -nearest neighbor learners

Brian M. Steele, Dept. of Mathematical Sciences, University of Montana, Missoula MT 59812

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 $\eta(X|\mathbf{Z})$ 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 of-

ten rival more sophisticated learners with respect to prediction error. The elementary k -nearest neighbor prediction of Y is $\eta(X|\mathbf{Z}) = k^{-1} \sum_{i=1}^k Y_{[i]}$, where $Z_{[i]} = (Y_{[i]}, X_{[i]})$ is the i th nearest training observation to a target $Z = (X, Y)$. Training observations are ordered according to distances between the target input X and the training sample inputs X_1, \dots, X_n . Classification problems involving c classes are accommodated by defining $Y = (y_1, \dots, y_c)$ where $y_j = 1$ if Z belongs to the j th class and otherwise $y_j = 0$. Then, the learner $\eta(X|\mathbf{Z})$ is an estimator of the class membership probabilities and the classification rule assigns Z to the class with the largest estimated probability of membership. Henceforth, quantitative target variables are assumed to be scalar, and so are denoted by y .

One approach to reducing the 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 goal of bagging is to construct a new learner that is the expectation of the original learner with respect to the empirical distribution function. To elucidate, let F_1 denote the empirical distribution function of \mathbf{Z} placing probability mass n^{-1} at each $Z_i \in \mathbf{Z}$ and 0 elsewhere, and \mathbf{Z}^* denote a random sample of n observations drawn with replacement from F_1 . The *exact* bagging learner is the expectation of $\eta(X|\mathbf{Z}^*)$ over F_1 , which is denoted herein as $\eta_1(X|\mathbf{Z}) = \mathbb{E}[\eta(X|\mathbf{Z}^*)|F_1]$. Generally, $\eta_1(X|\mathbf{Z})$ cannot be expressed analytically and a Monte Carlo algorithm is used to approximate $\eta_1(X|\mathbf{Z})$. Typically, B predictions are computed using bootstrap learners $\eta(X|\mathbf{Z}^{*1}), \dots, \eta(X|\mathbf{Z}^{*B})$ constructed from bootstrap samples $\mathbf{Z}^{*1}, \dots, \mathbf{Z}^{*B}$ drawn randomly and with replacement from F_1 . In the case of a quantitative target variable for example, the predictions are aggregated by averaging and the bagged prediction is $B^{-1} \sum_{b=1}^B \eta(X|\mathbf{Z}^{*b})$.

The application of bagging to k -nearest neighbor learners is unattractive from a com-

putational 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 $\eta_1(X|\mathbf{Z})$ 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 $Y = (y_1, y_2, \dots, y_c)$. Let $X_{1:n} = (X_{[1]}, X_{[2]}, \dots, X_{[n]})$ denote an ordered arrangement of 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 , given a metric on \mathbb{R}^p . In the examples below, Manhattan distance was used throughout. The ordering $X_{1:n}$ induces an ordering on \mathbf{Z} denoted by $Z_{1:n} = (Z_{[1]}, Z_{[2]}, \dots, Z_{[n]})$, where $Z_{[i]} = (X_{[i]}, Y_{[i]})$ (Bhattacharya 1974). The induced ordering on Y_1, Y_2, \dots, Y_n is denoted $Y_{1:n} = (Y_{[1]}, Y_{[2]}, \dots, Y_{[n]})$.

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

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

where $w_i = k^{-1}$ if $i \leq k$ and $w_i = 0$ if $k < i \leq n$. The second class of target 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 objective is to predict the class membership of Z from X . The output Y is a c -vector such that $y_j = 1$ if $Z \in \mathcal{P}_j$ and $y_j = 0$ if $Z \notin \mathcal{P}_j$. The posterior probability of membership in class \mathcal{P}_j is then $\pi_j = \Pr(Z \in \mathcal{P}_j|X) = \mathbb{E}(y_j|X)$. Herein, the k -nearest neighbor learner $\eta(X|\mathbf{Z})$ is an estimator of the c -vector of posterior probabilities $\pi(X) = [\pi_1(X), \dots, \pi_c(X)]$. A compact expression for $\eta(X|\mathbf{Z})$ is developed by setting

$$\mathbf{Y}_{1:n} = \begin{pmatrix} y_{[1],1} & \cdots & y_{[1],c} \\ \vdots & & \vdots \\ y_{[n],1} & \cdots & y_{[n],c} \end{pmatrix} \quad (1)$$

where the i th row of $\mathbf{Y}_{1:n}$ is $Y_{[i]}$. Then, $\eta(X|\mathbf{Z}) = w^T \mathbf{Y}_{1:n} = \hat{\pi}(X|\mathbf{Z})$ is an estimator of $\pi(X)$ and $\hat{\pi}_j(X|\mathbf{Z})$ is the proportion of the k -nearest neighbors of Z belonging to class \mathcal{P}_j . The k -nearest neighbor *classifier* is $\arg \max_j \hat{\pi}_j(X|\mathbf{Z})$. In the case of a tie among the largest values of $\hat{\pi}_1(X|\mathbf{Z}), \dots, \hat{\pi}_c(X|\mathbf{Z})$, the neighborhood size k may be successively increased until the tie is broken.

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 prediction of Y is

$$\widehat{\eta}_1(X|\mathbf{Z}) = B^{-1} \sum_{b=1}^B \eta(X|\mathbf{Z}^{*b}). \quad (2)$$

If the bagged learner is used for classification, then the class membership of Z may be predicted using voting scheme in which each bootstrap learner produces a prediction and the class most frequently predicted is taken to be the ensemble prediction. Alternatively, class membership may be predicted by $\arg \max_j \widehat{\eta}_{1,j}(X|\mathbf{Z})$.

Let $\mathbf{Z}^* = (Z_1^*, \dots, Z_n^*)$ denote a bootstrap sample drawn randomly from F_1 . The bagged learner given in formula (2) is an estimator of the exact bootstrap expectation

$$\eta_1(X|\mathbf{Z}) = \mathbb{E}[\eta(X|\mathbf{Z}^*)|F_1] = n^{-n} \sum_{i \in \mathcal{I}} \eta(\mathbf{Z}^i|X) \quad (3)$$

where \mathcal{I} is the set of all n -tuples formed by choosing n integers with replacement from $\{1, \dots, n\}$ and $\mathbf{Z}^i = (Z_{i_1}, \dots, Z_{i_n})$ is an n -tuple of elements drawn from \mathbf{Z} and indexed by the elements comprising $i = (i_1, \dots, i_n)$. 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.

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

Consider first a quantitative (and scalar) output. For some $X \in \mathcal{P}$, let $Y_{1:n}^* = (y_{[1]}^*, \dots, y_{[n]}^*)$ denote the order statistic induced by the distances between X and X_i^* , $i = 1, \dots, n$. The exact bootstrap expectation of the k -nearest neighbor learner is

$$\eta_1(X|\mathbf{Z}) = \mathbb{E}(w^T Y_{1:n}^* | F_1).$$

If the output variable is multinomial and therefore identifying class membership, the induced order statistic is denoted by the $n \times c$ matrix $\mathbf{Y}_{1:n}^*$ analogous to equation (1) and the exact bootstrap expectation of $\eta(X|\mathbf{Z})$ is $\eta_1(X|\mathbf{Z}) = w^T \mathbb{E}(\mathbf{Y}_{1:n}^* | F_1)$.

To develop an analytic formula for the exact bootstrap expectation $E(y_{[i]}^*|F_1)$, or more generally $E(Y_{[i]}^*|F_1)$, note that $Y_{[i]}^*$ is the output associated with the i th nearest neighbor $Z_{[i]}^*$. The only possible realization of $Y_{[i]}^*$ is one of $Y_{[1]}, \dots, Y_{[n]}$, and $Y_{[i]}^*$ will be $Y_{[j]}$ if and only if $X_{[i]}^* = X_{[j]}$ and equivalently, $Z_{[i]}^* = Z_{[j]}$. Consequently,

$$E(Y_{[i]}^*|F_1) = \sum_{j=1}^n \Pr(Z_{[i]}^* = Z_{[j]}|F_1)Y_{[j]}.$$

The computation of $\Pr(Z_{[i]}^* = Z_{[j]}|F_1)$ is addressed momentarily. First, let $\mathbf{P} = (\mathbf{P}_{ij})$ denote the $n \times n$ matrix with $\mathbf{P}_{ij} = \Pr(Z_{[i]}^* = Z_{[j]}|F_1)$ in the i th row and j th column. Then $E(Y_{1:n}^*|F_1) = \mathbf{P}Y_{1:n}$ and $\eta_1(X|\mathbf{Z}) = w^T E(Y_{1:n}^*|F_1) = w^T \mathbf{P}Y_{1:n}$. In the case of the elementary k -nearest neighbor learner with $w_i = k^{-1}$ if $i \leq k$ and $w_i = 0$ if $i > k$,

$$\eta_1(X|\mathbf{Z}) = w^T \mathbf{P}Y_{1:n} = k^{-1} \sum_{j=1}^n \sum_{i=1}^k \Pr(Z_{[i]}^* = Z_{[j]}|F_1)Y_{[j]}. \quad (4)$$

From equation (4), it is apparent that $\eta_1(X|\mathbf{Z})$ is a weighted mean of $Y_{[1]}, \dots, Y_{[n]}$ where the weight associated with the j th nearest neighbor $Y_{[j]}$ is $k^{-1} \sum_{i=1}^k \Pr(Z_{[i]}^* = Z_{[j]}|F_1)$.

3.1.1 An analytic expression for computing \mathbf{P}_{ij}

The bootstrap probabilities $\mathbf{P}_{ij} = \Pr(Z_{[i]}^* = Z_{[j]}|F_1)$ can be derived easily using properties of the binomial distribution. Let $\text{Bin}(n, j/n)$ denote the binomial distribution with binomial denominator n and probability j/n ; let E_{ij}^* denote the event that a bootstrap sample \mathbf{Z}^* contains at least i elements selected from $\mathbf{Z}_{1:j} = \{Z_{[1]}, \dots, Z_{[j]}\}$, and S_j^* denote the number of elements in \mathbf{Z}^* belonging to $\mathbf{Z}_{1:j}$. Note that $\Pr(E_{ij}^*|F_1) = \Pr(S_j^* \geq i|F_1)$ and $S_j^* \sim \text{Bin}(n, j/n)$ because the elements of the bootstrap sample are drawn independently and with replacement from F_1 . Let \overline{E} denote the complement of the event E ; then $E_{ij}^* \cap \overline{E_{i,j-1}^*}$ is the event that at least i elements are selected from $\mathbf{Z}_{1:j}$ and less than i elements are selected from $\mathbf{Z}_{1:j-1}$. Thus, $E_{ij}^* \cap \overline{E_{i,j-1}^*}$ will occur if and only if $Z_{[i]}^* = Z_{[j]}$

occurs. Since $E_{i,j-1}^* \subset E_{ij}^*$,

$$\begin{aligned} \Pr(Z_{[i]}^* = Z_{[j]}|F_1) &= \Pr(E_{ij}^* \cap \overline{E}_{i,j-1}^*) \\ &= \Pr(E_{ij}^*) - \Pr(E_{i,j-1}^*) \\ &= \Pr(S_j^* \geq i) - \Pr(S_{j-1}^* \geq i). \end{aligned}$$

Computing $\Pr(Z_{[i]}^* = Z_{[j]}|F_1)$ is straightforward because $\Pr(S_j^* \geq i)$ is a sum of $n - i + 1$ binomial probabilities. A well-known relationship between the binomial and beta distributions (Mood et al.1974) can be exploited to simplify and speed up computation. Specifically, $\Pr(S_j^* \geq i) = F_{i,n-i+1}(j/n)$, where $F_{\alpha,\beta}(x)$ is the cumulative distribution function of a beta random variable with parameters α and β evaluated at x . Thus,

$$\Pr(Z_{[i]}^* = Z_{[j]}|F_1) = F_{i,n-i+1}(j/n) - F_{i,n-i+1}[(j-1)/n]. \quad (5)$$

The effort of computing the exact bagging k -nearest neighbor prediction $\eta_1(X|\mathbf{Z}) = w^T \mathbf{P}Y_{1:n}$ differs little from that of the elementary k -nearest neighbor because the $n \times n$ matrix $\mathbf{P} = (\mathbf{P}_{ij})$ depends only on the number of training observations. Hence, \mathbf{P} need only be computed once before $\eta_1(\cdot|\mathbf{Z})$ is used for prediction.

Hutson and Ernst (2000) present more general computational 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, \dots, y_n but is instead induced by the distances between the target input X and the training sample inputs X_1, \dots, X_n . Hutson and Ernst's (2000) formula for the bootstrap probability $\Pr(Z_{[i]}^* = Z_{[j]}|F_1)$ is equivalent to formula (5), though their derivation is quite different from that presented above. Caprile et al. (2004) and Steele et al. (2003) have derived other formulae for $E[\eta(X|\mathbf{Z}^*)|F_1]$. The computational demands of these formulae are substantially greater than formula (5) and Hutson and Ernst's (2000) formula.

3.2 Sub-sampling

Bootstrap sub-aggregation is carried out by sampling $m < n$ observations randomly from F_1 (Bickel et al. 1997; Bühlmann and Yu 2002; Hall and Samworth 2005). For the bagged *nearest* neighbor classifier ($k = 1$), Hall and Samworth (2005) have presented asymptotic arguments and practical examples 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.

Suppose that $m < n$ observations are sampled randomly and *with replacement* from F_1 . Let $\mathbf{Z}_{1:m}^* = (Z_{[1]}^*, \dots, Z_{[m]}^*)$ denote the ordered m -tuple comprised of the m nearest neighbors to a target Z among a bootstrap sample drawn from F_1 . The vector of weights w under sub-sampling consists of m coefficients $w_i = k^{-1}$ if $1 \leq i \leq k$ and $w_i = 0$ if $k < i \leq m$, and the learner constructed from $\mathbf{Z}_{1:m}^*$ is $\eta(X|\mathbf{Z}^*) = w^T Y_{1:m}^*$. The exact bootstrap sub-aggregated k -nearest neighbor learner is $\eta_1(X|\mathbf{Z}) = w^T \mathbb{E}(Y_{1:m}^* | F_1)$, where $\mathbb{E}(Y_{[i]}^* | F_1) = \sum_{j=1}^n \Pr(Z_{[i]}^* = Z_{[j]} | F_1) Y_{[j]}$, $i = 1, \dots, m$. To derive a formula for $\Pr(Z_{[i]}^* = Z_{[j]} | F_1)$, let E_{ij}^* denote the event that a bootstrap sample $\mathbf{Z}_{1:m}^*$ contains at least i elements selected from $\mathbf{Z}_{1:j} = \{Z_{[1]}, \dots, Z_{[j]}\}$ and S_j^* denote the number of elements in $\mathbf{Z}_{1:m}^*$ belonging to $\mathbf{Z}_{1:j}$. As before, $\Pr(E_{ij}^* | F_1) = \Pr(S_j^* \geq i | F_1)$ and $E_{ij}^* \cap \overline{E_{i,j-1}^*}$ will occur if and only if $Z_{[i]}^* = Z_{[j]}$ occurs. However, because $m < n$ observations are sampled, $S_j^* \sim \text{Bin}(m, j/n)$ and

$$\begin{aligned} \Pr(Z_{[i]}^* = Z_{[j]} | F_1) &= \Pr(S_j^* \geq i) - \Pr(S_{j-1}^* \geq i) \\ &= F_{i, m-i+1}(j/n) - F_{i, m-i+1}[(j-1)/n], \end{aligned} \tag{6}$$

for $1 \leq i \leq m$ and $j = 1, \dots, n$. For $i > m$, $\Pr(Z_{[i]}^* = Z_{[j]} | F_1) = 0$, $j = 1, \dots, n$.

To express the exact bagging k -nearest learner as a linear function of $Y_{1:n}$, the $m \times n$ matrix $\mathbf{P} = (\mathbf{P}_{ij})$ is again defined by setting $\mathbf{P}_{ij} = \Pr(Z_{[i]}^* = Z_{[j]} | F_1)$. It follows that the exact bagging k -nearest neighbor learner under sub-sampling with replacement is $\eta_1(X|\mathbf{Z}) = w^T \mathbf{P} Y_{1:n}$.

Now suppose that sampling is *without replacement*. If $j < i$, then $\Pr(Z_{[i]}^* = Z_{[j]}) = 0$. Suppose instead $j \geq i$; 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 $j \geq i$,

$$\Pr(Z_{[i]}^* = Z_{[j]} | F_1) = \frac{\binom{j-1}{i-1} \binom{n-j}{m-i}}{\binom{n}{m}}. \quad (7)$$

As before, defining $\mathbf{P}_{ij} = \Pr(Z_{[i]}^* = Z_{[j]} | F_1)$ leads to $\eta_1(X|\mathbf{Z}) = w^T \mathbf{P} Y_{1:n}$.

3.3 k -nearest neighbor weights

The exact bagging k -nearest neighbor learner predicts a target Y by a weighted mean of the ordered training outputs. Specifically, $\eta_1(X|\mathbf{Z}) = w^T \mathbf{P} Y_{1:n}$ so that the row vector of weights is $c = w^T \mathbf{P}$. As shown in formula (4), the j th weight is $c_j = k^{-1} \sum_{i=1}^k \Pr(Z_{[i]}^* = Z_{[j]} | F_1)$. Differences among weights as a function of k and sampling scheme are illustrated in Figures 1 and 2. Figure 1 plots the weights c_j against j when sampling without replacement and for $n = 20$ for $k \in \{1, 3, 5, 8\}$. Figure 2 is the same as Figure 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 Figures 1 and 2. Figures 1 and 2 show that the weight c_j associated with $Z_{[j]}$ depends on j , k and the bootstrap sampling scheme. For all m , k and j , the exact bagging weight c_j satisfies $0 < c_j < k^{-1}$ in contrast to the elementary k -nearest neighbor weights $w_j \in \{0, k^{-1}\}$. 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, Brieman 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.

4 k -nearest neighbor regression learners

Suppose that the target variable 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 \mathbb{R}^p instead of being globally constant. A varying linear function is accommodated within the k -nearest neighbor framework by letting Z_0 denote the target and supposing that $E(y_0|X_0) = X_0^T \beta_0$ where β_0 is an unknown vector of coefficients. Suppose further that $E(y_{[j]}|X_{[j]}) = X_{[j]}^T \beta_0$ holds for $j = 1 \dots, k$. If the linear model is a reasonable approximation of the relationship between $E(y_{[j]}|X_{[j]})$ and $X_{[j]}$, $j = 1, \dots, k$, then a least squares approach to estimating β_0 and predicting y_0 may be fruitful. To proceed, let $\psi_0(Z)$ denote the indicator function of the event $Z \in \{Z_{[1]}, \dots, Z_{[k]}\}$ and Ψ_0 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 as before 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

$$S(\beta_0|\mathbf{Z}) = (Y - \mathbf{X}\beta_0)^T \Psi_0 (Y - \mathbf{X}\beta_0) \quad (8)$$

with respect to β_0 . Provided that $\mathbf{X}^T \boldsymbol{\Psi}_0 \mathbf{X}$ is nonsingular and the locally constant model $E(y_{[j]}|X_{[j]}) = X_{[j]}^T \beta_0$ $j = 1, \dots, k$, is correct, then the least squares estimator of β_0 is $\hat{\beta}_0 = (\mathbf{X}^T \boldsymbol{\Psi}_0 \mathbf{X})^{-1} \mathbf{X}^T \boldsymbol{\Psi}_0 Y$. The k -nearest neighbor regression learner predicts y_0 by $\eta(X_0|\mathbf{Z}) = X_0^T \hat{\beta}_0$. When k is small, the variance of $\hat{\beta}_0$ may be large and the learner unstable; worse, when $k \approx p$, $\mathbf{X}^T \boldsymbol{\Psi}_0 \mathbf{X}$ 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 the target 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 $k \leq 20$. 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 $\tilde{\beta} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T Y$ towards 0, and the effect on the learner is to reduce instability. A regularized local estimator $\tilde{\beta}_0 = (\mathbf{X}^T \boldsymbol{\Psi}_0 \mathbf{X} + 10^{-5} \mathbf{I})^{-1} \mathbf{X}^T \boldsymbol{\Psi}_0 Y$ 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 $a_i = X^T (\mathbf{X}^T \boldsymbol{\Psi}_Z \mathbf{X})^{-1} \mathbf{X}^T \boldsymbol{\Psi}_i$ where $\boldsymbol{\Psi}_i$ is the i th column of $\boldsymbol{\Psi}_0$. Though the k -nearest neighbor regression learner is a linear combination of $Y_{1:n}$, this learner does not satisfy the conditions necessary for $E(aY_{1:n}^*|F_1) = aE(Y_{1:n}^*|F_i)$ 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 appears not to 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 $E[S(\beta_0|\mathbf{Z})|F_1]$ [formula (8)]. 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 $\beta_0^E = E(\mathbf{X}^{*T} \Psi_0^* \mathbf{X}^* | F_1)^{-1} E(\mathbf{X}^{*T} \Psi_0^* Y^* | F_1)$ minimizes $E[S(\beta_0 | \mathbf{Z}) | F_1] = E\{(Y^* - \mathbf{X}^* \beta_0)^T \Psi_0^* (Y^* - \mathbf{X}^* \beta_0) | F_1\}$.*

See the Appendix for the proof. Theorem 2 establishes that $E(\mathbf{X}^{*T} \Psi_0^* \mathbf{X}^* | F_1)^{-1}$ and $E(\mathbf{X}^{*T} \Psi_0^* Y^* | F_1)$ are computationally simple.

Theorem 2. *Without loss of generality, assume that the rows of \mathbf{X} , Y , and Ψ_0 have been arranged in ascending order according to the distances between X_0 and X_1, \dots, X_n . Let \mathbf{A} denote a diagonal matrix such that the j th diagonal element is $\sum_{i=1}^k \Pr(Z_{[i]}^* = Z_{[j]} | F_1)$. Then, $E(\mathbf{X}^{*T} \Psi_0^* Y^* | F_1) = \mathbf{X}^T \mathbf{A} Y$ and $E(\mathbf{X}^{*T} \Psi_0^* \mathbf{X}^* | F_1) = \mathbf{X}^T \mathbf{A} \mathbf{X}$. Furthermore, $\mathbf{X}^T \mathbf{A} \mathbf{X}$ is full rank and $\beta_0^E = (\mathbf{X}^T \mathbf{A} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{A} Y$.*

See the Appendix for the proof. In principle, replacing $\mathbf{X}^T \Psi_0 \mathbf{X}$ by $E(\mathbf{X}^{*T} \Psi_0^* \mathbf{X}^* | F_1) = \mathbf{X}^T \mathbf{A} \mathbf{X}$ 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^E 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 summarize the data sets. Manhattan distance was used in all application of the k -nearest neighbor learners in this study. 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 as a training set of $[.9n]$ observations and a test set of $n - [.9n]$ 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.

The k -nearest neighbor learners were constructed for $k \in \mathcal{K} = \{1, 5, 10, 20, 50\}$. 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 $r \in \mathcal{R} = \{.9, .8, .7, .6, .5\}$.

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 $k \in \mathcal{K}$; 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 $k \in \mathcal{K}$ and $r \in \mathcal{R}$. 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 consistently smaller error estimates than the competitors. Table 3 also shows that exact bagging k -nearest neighbor learners, including those than 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 estimates of error do not consistently favor bagging. Similarly, the effectiveness of sub-sampling for specific k varied without consistency among data sets.

A closer look at exact bagging compares error estimates produced by the elementary and the exact bagging versions of the k -nearest neighbor learners for each value of k . Some scaling of the estimates is helpful, and this was accomplished by letting $e_0(k, \mathbf{Z})$ and $e_1(k, \mathbf{Z})$ denote cross-validation estimator of prediction error for an elementary k -nearest neighbor learner and an exact bagging counterpart, respectively. For the classification problems, the difference in prediction error $d(k, \mathbf{Z}) = e_0(k, \mathbf{Z}) - e_1(k, \mathbf{Z})$ was plotted against the scaled prediction error

$$s_0(k, \mathbf{Z}) = \frac{e_0(k, \mathbf{Z}) - \min_j e_0(j, \mathbf{Z})}{\max_j e_0(j, \mathbf{Z}) - \min_j e_0(j, \mathbf{Z})}.$$

in Figure 3. Fifty comparisons are obtained from the 10 data sets involving categorical outputs (since there were 5 choices of k). Figure 3 shows that the exact bagging learner prediction error estimate was smaller than or equal to the elementary learner prediction error for 25 of the 50 comparisons and the estimates were equal for 11 comparisons. However, it ought to be kept in mind that when prediction error for the elementary k -nearest neighbor learner was a minimum (over $k \in \mathcal{K}$) for a specific data set, then the difference in error estimates was zero (to four significant digits) for 5 of the 10 data sets. A slightly different approach was taken when comparing estimates for quantitative outputs. In this case, $\ln[e_0(k, \mathbf{Z})]$ was plotted against $e_1(k, \mathbf{Z})$ where $e_1(k, \mathbf{Z})$ is the estimated error derived from the partially exact bagging k -nearest neighbor regression learner and $e_0(k, \mathbf{Z})$ is the error estimate derived from the regularized k -nearest neighbor regression learner. The natural logarithm transformation was adopted to reduce differences between data sets and hence improve the clarity of Figure 4. Of the 35 comparisons shown in Figure 4, the

partially exact regression learner yielded a smaller estimate of prediction error 21 times and the estimates were equal 9 times. As with the classification problems, the smallest error estimates for a specific data set were often indistinguishable. Finally, a comparison of prediction error estimates obtained from the elementary and exact bagging k -nearest neighbor learners revealed that 27 of 35 exact bagging estimates were smaller than the elementary k -nearest neighbor learners (figure not shown). Two of the estimates were tied. 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. In summary, for less than optimal choices of k , bagging often improves on the conventional k -nearest neighbor learners, but when k is selected by examining cross-validation estimates of error, then the differences in prediction error largely disappear.

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 shrinks 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 extent of shrinking 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.

7 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.
- 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: Roli, F., Kittler, J., & Windeatt, T. (Eds.): MCS 2004, LNCS 3077, 72-81, Berlin: Springer-Verlag.
- 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.
- 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,
- 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.
- Kelly, D.G. (1994). *Introduction to Probability*. New York: Macmillan Publishing Co.

- 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*, 546-551. Providence, RI.
- Mood, A.M., Graybill, F.A., & Boes, D.C. (1974). *An Introduction to the Theory of Statistics*. New York: Wiley.
- Ripley, B.D. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.
- Ruppert, D., Wand, M.P. & Carroll, R.J. (2003). *Semiparametric Regression*. 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. (2000). Ideal bootstrap estimation of expected prediction error for k -nearest neighbor classifiers: Applications for classification and error assessment. *Statistics and Computing*, 10, 349-355.
- Wolpert, D.H. (1992). Stacked generalization. *Neural Networks*, 5(2), 241-259.

A Appendix: Proofs

Proof of Theorem 1

The order of differentiation and integration can be reversed when differentiating $E[S(\beta_0|\mathbf{Z}^*)|F_1]$ with respect to β_0 because $S(\beta_0|\mathbf{Z}^*)$ [equation (8)] is a continuous function of β_0 and F_1 is a discrete distribution function with at most n^n points with non-zero probability. Hence

$$\frac{\partial E[S(\beta_0|\mathbf{Z}^*)|F_1]}{\partial \beta_0} = -2E[\mathbf{X}^{*T}\Psi_0^*(Y^* - \mathbf{X}^*\beta_0)|F_1]. \quad (9)$$

Setting the vector of partial derivatives equal to 0 yields the normal equations $E(\mathbf{X}^{*T}\Psi_0^*\mathbf{X}^*|F_1) = E(\mathbf{X}^{*T}\Psi_0^*Y^*|F_1)\beta_0$. Suppose now that the $n \times p$ design matrix \mathbf{X} is full rank. Theorem 2 shows that $E(\mathbf{X}^{*T}\Psi_0^*\mathbf{X}^*|F_1)$ is positive definite and consequently the unique solution to the normal equations is

$$\beta_0^E = E(\mathbf{X}^{*T}\Psi_0^*\mathbf{X}^*|F_1)^{-1}E(\mathbf{X}^{*T}\Psi_0^*Y^*|F_1). \quad (10)$$

Furthermore, since $E(\mathbf{X}^{*T}\Psi_0^*\mathbf{X}^*|F_1)$ is positive definite,

$$\frac{\partial^2 E[S(\beta_0|\mathbf{Z}^*)|F_1]}{\partial \beta \partial \beta^T} = -E(\mathbf{X}^{*T}\Psi_0^*\mathbf{X}^*|F_1) \quad (11)$$

is negative definite and it follows that β_0^E minimizes $E[S(\beta_0|\mathbf{Z}^*)|F_1]$.

Proof of Theorem 2 Let $X_j = (x_{j,1}, \dots, x_{n,j})^T$ denote the j th column of \mathbf{X} . Note that the r th diagonal element of Ψ_0 indicates the event $\{r \leq k\}$. Then, the i, j th element of $E(\mathbf{X}^{*T}\Psi_0^*\mathbf{X}^*|F_1)$ is

$$\begin{aligned} E(\mathbf{X}_i^{*T}\Psi_0^*\mathbf{X}_j^*|F_1) &= \sum_{r=1}^k E(x_{[r],i}^*x_{[r],j}^*|F_1) \\ &= \sum_{r=1}^k \sum_{s=1}^n x_{[s],i}x_{[s],j} \Pr(Z_{[r]}^* = Z_{[s]}|F_1) \\ &= \sum_{s=1}^n x_{[s],i}x_{[s],j} \sum_{r=1}^k \Pr(Z_{[r]}^* = Z_{[s]}|F_1) \\ &= X_i^T \mathbf{A} X_j, \end{aligned}$$

because \mathbf{A} is diagonal and the r th diagonal element of \mathbf{A} is defined to be $\sum_{r=1}^k \Pr(Z_{[r]}^* = Z_{[s]}|F_1)$. Hence, $E(\mathbf{X}^{*T} \Psi_0^* \mathbf{X}^* | F_1) = \mathbf{X}^T \mathbf{A} \mathbf{X}$.

The calculation of $E(\mathbf{X}^{*T} \Psi_0^* Y^* | F_1)^T = [E(X_1^{*T} \Psi_0^* Y^* | F_1), \dots, E(X_p^{*T} \Psi_0^* Y^* | F_1)]$ proceeds in the same fashion. The j th element is

$$\begin{aligned} E(X_j^{*T} \Psi_0^* Y^* | F_1) &= \sum_{r=1}^k E(x_{[r],j}^* y_{[r]}^* | F_1) \\ &= \sum_{s=1}^n x_{[s]j} y_{[s]} \sum_{r=1}^k \Pr(Z_{[r]}^* = Z_{[s]} | F_1) \\ &= X_j^T \mathbf{A} Y. \end{aligned}$$

Thus, $E(\mathbf{X}^{*T} \Psi_0^* Y^* | F_1) = \mathbf{X}^T \mathbf{A} Y$.

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 $\sum_{r=1}^k \Pr(Z_{[r]}^* = Z_{[s]} | F_1) > 0$. Hence, $\text{rank}(\mathbf{X}^T \mathbf{A} \mathbf{X}) = \text{rank}(\mathbf{X}^T \mathbf{X})$. Thus, $\mathbf{X}^T \mathbf{A} \mathbf{X}$ is full rank and nonsingular.

Tables

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	$[\.9n]$	100	9	2
Diabetes	768	$[\.9n]$	100	8	2
Ecoli	336	$[\.9n]$	100	7	8
Glass	214	$[\.9n]$	100	9	6
Image	2310	$[\.9n]$	100	19	7
Ionosphere	315	$[\.9n]$	100	34	2
Satellite image	6435	4435	1	36	6
Sonar	208	$[\.9n]$	100	60	2
Vehicle	846	$[\.9n]$	100	18	4
Vowel	990	$[\.9n]$	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	$[\.75n]$	10	8
Boston housing	506	$[\.9n]$	100	12
Friedman #1	2200	200	1	10
Friedman #2	2200	200	1	4
Friedman #3	2200	200	1	4
Ozone	330	$[\.9n]$	100	8
Servo	167	$[\.9n]$	100	4

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.

Set	k -NN	Exact bagging	Sub-sampling		Adaboost	Random Forest
			with	without		
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 $k \in \{1, 5, 10, 20\}$. The minimum prediction error estimates among all values of k are presented below.

Set	k -NN	Exact bagging	Reg. k -NN regression	PEB k -NN regression			Adaptive bagging	Random Forest
				none	with	without		
Abalone	4.99	4.95	4.69	4.63	4.95	4.47	4.9	4.6
Boston housing	20.8	16.4	13.1	13.1	16.3	12.7	9.7	10.2
Ozone	9.95	9.85	8.92	8.97	9.94	8.79	17.8	16.3
Servo	.698	.703	.456	.455	.724	.401	.251	.246
Friedman #1	9.09	8.70	5.89	5.87	8.74	6.09	4.1	5.7
Friedman # 2×10^3	32.9	32.3	20.5	20.5	33.9	20.0	21.5	19.6
Friedman # 3×10^{-3}	39.1	37.9	33.0	32.0	39.8	30.5	24.8	21.6

Figures

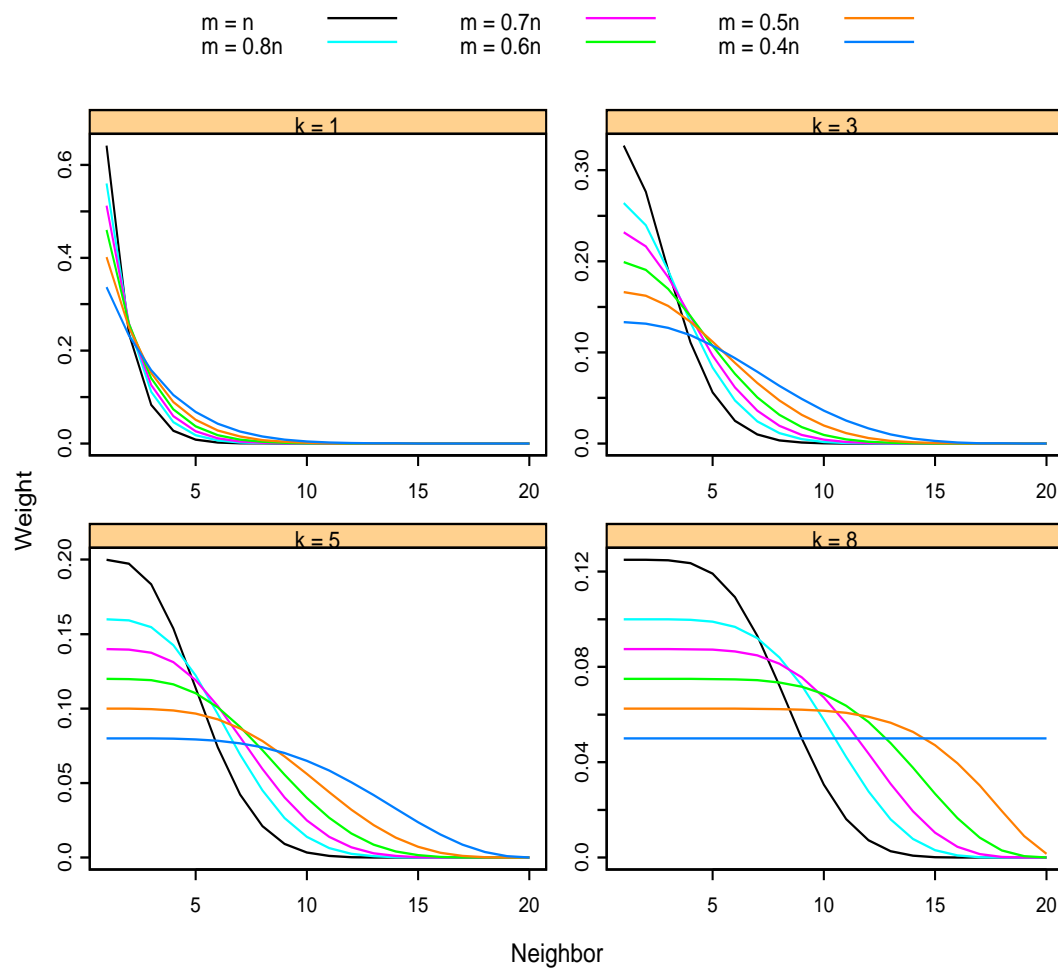


Figure 1: Exact bootstrap weights for each neighbor given $k \in \{1, 3, 5, 8\}$, sub-sample size m , and a training sample size of $n = 20$. Note that the vertical scale differs among panels.

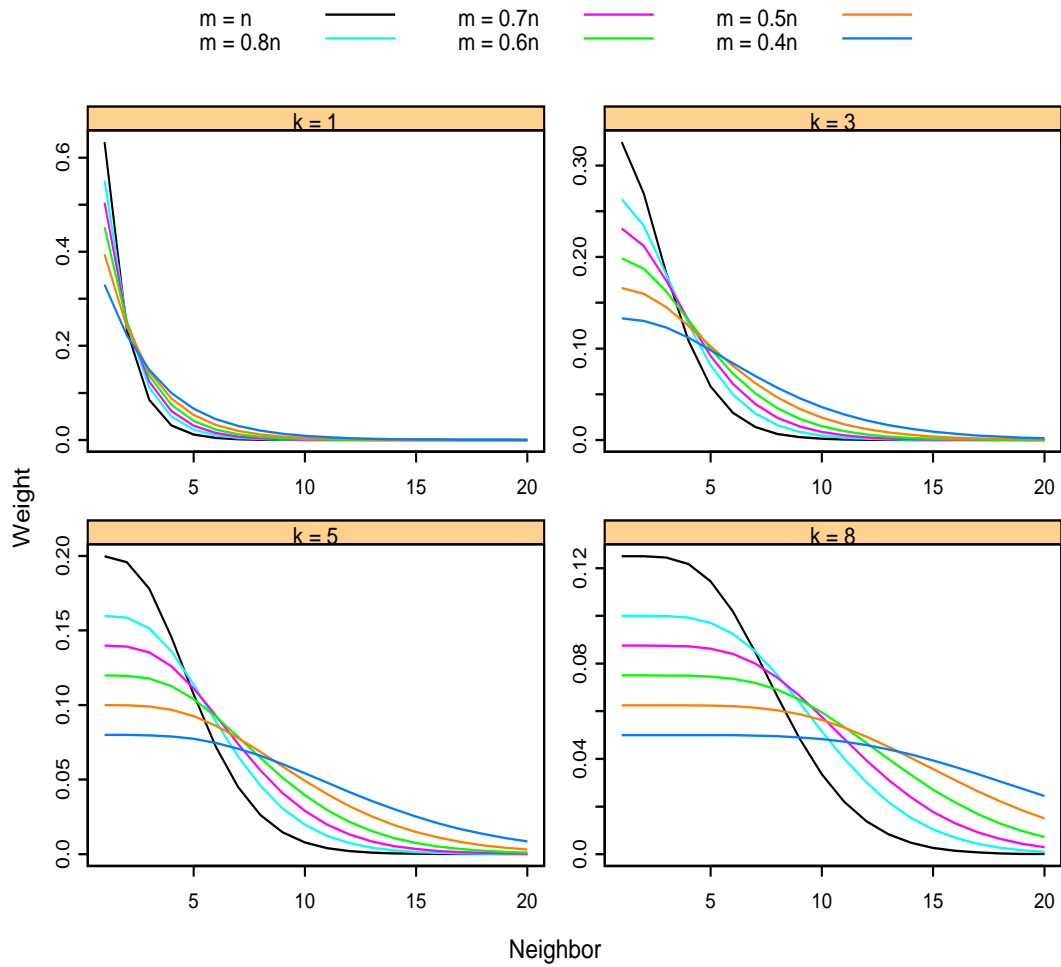


Figure 2: Exact bootstrap weights for the nearest 20 neighbors for $k \in \{1, 3, 5, 8\}$, sub-sample size m , and a training sample size of $n = 200$. Note that the vertical scale differs among panels.

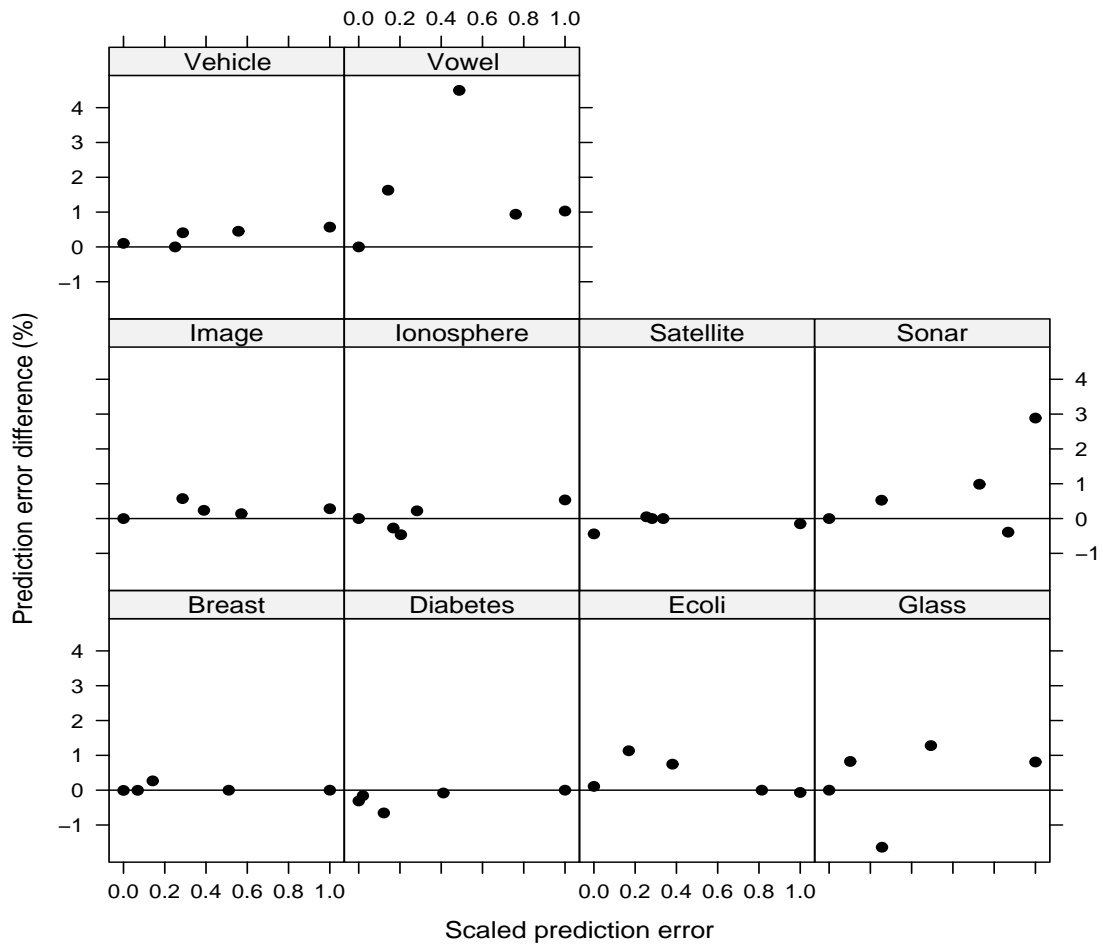


Figure 3: Categorical output variables. Prediction error difference plotted against scaled elementary k -nearest neighbor learner prediction error for $k \in \{1, 5, 10, 20, 50\}$.

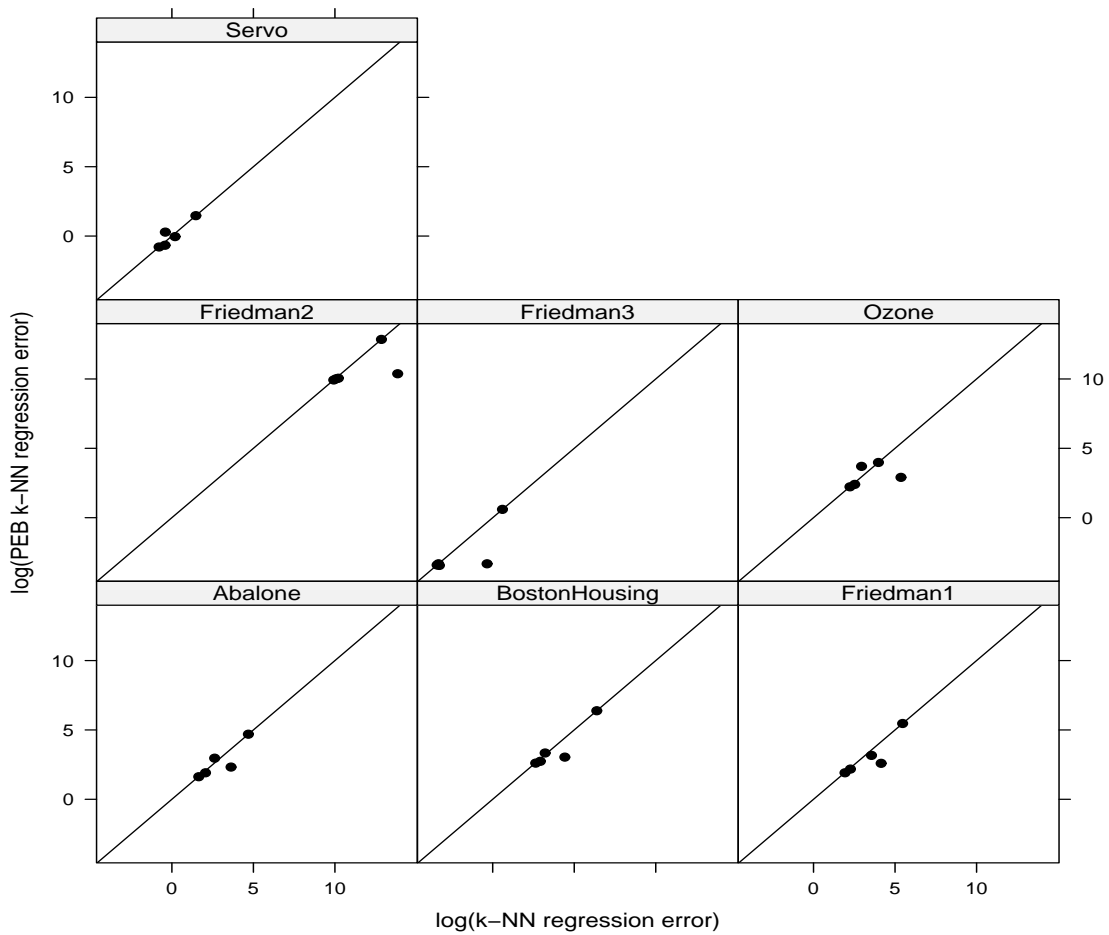


Figure 4: Prediction error difference plotted against scaled regularized k -nearest neighbor regression learner prediction error for $k \in \{1, 5, 10, 20, 50\}$. Also shown is a line with slope 1 and intercept 0.