Bias, variance and prediction error for classification rules

Robert Tibshirani *
Department of Preventive Medicine and Biostatistics
and Department of Statistics
University of Toronto
Toronto, Canada

November 6, 1996

Abstract

We study the notions of bias and variance for classification rules. Following Efron (1978) we develop a decomposition of prediction error into its natural components. Then we derive bootstrap estimates of these components and illustrate how they can be used to describe the error behaviour of a classifier in practice. In the process we also obtain a bootstrap estimate of the error of a "bagged" classifier.

Keywords: classification, prediction error, bias, variance, bootstrap

1 Introduction

This article concerns classification rules that have been constructed from a set of training data. The training set $\mathcal{X} = (x_1, x_2, \cdots, x_n)$ consists of n observations $x_i = (t_i, g_i)$, with t_i being the predictor or feature vector and g_i being the response, taking values in $\{1, 2, \ldots K\}$. On the basis of $\mathcal{X}$ the

*Addresses: tibs@utstat.toronto.edu; http://www.utstat.toronto.edu/~tibs

statistician constructs a classification rule $C(t, \mathcal{X})$. Our objective here is to understand the bias, variance, and prediction error of $C(t, \mathcal{X})$.

For convenience we define $y = (y^1, \ldots y^K)$ to be the indicator-variable coding of g. Specifically, let e_k be a K-vector of zeroes, except for a one in the kth position. Then $y = e_k$ if $g = k$. We denote the output of the classifier $C(t, \mathcal{X})$ by the vector $(c^1, \ldots c^K)$, having elements in $[0, 1]$ adding up to one. This output may be a vector in $\{e_1, e_2, \ldots e_K\}$ or may be a set of probabilities. For simplicity we assume that the cost of misclassifying class j as class $k \neq j$ is the same for all j, k.

We use $Q[y, c]$ to indicate the loss function between a predicted value c and the actual response y. Both the arguments y and c are K-vectors with elements in $[0, 1]$ and summing up to one. They can be vectors in $\{e_1, e_2, \ldots e_K\}$, but need not be. We allow the first argument to be any probability vector, for the definitions introduced later in the section.

The choice of Q plays an important role in defining bias, variance and prediction error for the classifier C. One choice for Q of particular interest is

$$Q_1[y, c] \;\; = \;\; y^a - y^b \tag{1}$$

where $a = \text{argmax}(y), b = \text{argmax}(c)$. This loss is the difference in probability between the highest probability class and the class predicted by the rule c. Note that when the y takes values only 0 or 1, $Q_1[y, c]$, counts classification errors in c. The value of Q_1 is the same whether c is a vector of probabilities, or is converted to a vector of zeroes, with a one in the position corresponding to the class with the highest probability. This definition generalizes the two-class definition given in Efron (1978), and must be refined to handle the case where the maximum of c is not unique: see section 6.

Other popular choices for Q are squared error

$$Q_2[y, c] = \sum_k (y^k - c^k)^2 \tag{2}$$

and multinomial deviance or cross-entropy

$$Q_3[y, c] = -2 \sum_k y^k \log c^k. \tag{3}$$

These are summarized in Table 1 and discussed further in Section 6.

Table 1: *Some choices for the loss function $Q[y,c]$. $\sigma(p)$ is the dispersion function, discussed in section 6.*

	Name	$Q[y,c]$	$\sigma(p)$
1.	Misclassification error	$y^a - y^b$, $a = \text{argmax}(y), b = \text{argmax}(c)$	$1 - \max(p)$
2.	Squared error	$\sum_k (y^k - c^k)^2$	$\sum_{j \neq k} p^j p^k$
3.	Multinomial deviance/ Cross-entropy	$-2 \sum_k y^k \log c^k$	$-2 \sum_k p^k \log p^k$

We assume that the observations $x_i = (t_i, y_i)$ in the training set are a random sample from some distribution F,

$$x_1, x_2, \cdots, x_n \overset{\text{i.i.d.}}{\sim} F, \tag{4}$$

and that $x = (t, y)$ is another independent draw from F, called a test point. The prediction error of the rule $C(t, \mathcal{X})$ is defined as

$$\text{PE}(Y, C) \equiv \text{E}_F \text{E}_{0F} Q[Y, C(t, \mathcal{X})] \tag{5}$$

Here E_F refers to expectation over the training set $\mathcal{X}$ whose members are i.i.d. F, and E_{0F} refers to expectation over the test observation $x = (t, y) \sim F$. This is called the expected error rate in Efron & Tibshirani (1995), distinct from the conditional error rate which fixes the training set $\mathcal{X}$. While the conditional error rate might often be of main interest, studies in Efron & Tibshirani (1995) show that it is very difficult to estimate. Hence one usually focusses on the unconditional error rate.

In the regression problem with squared error loss, a simple decomposition exists for PE. If Y is a continuous-valued random variable we let

$$Y = C_0(t) + \epsilon \tag{6}$$

where $\mathrm{E}(\epsilon|t) = 0$. Let $\sigma^2(t) = \mathrm{Var}(Y|t)$. Then for an estimator $C(t, \mathcal{X})$ with $C_A(t) = \mathrm{E}_F[C(t, \mathcal{X})]$, we have

$$\begin{aligned} \mathrm{Bias}(C) &= \mathrm{E}_{0F}[C_A(t) - C_0(t)]^2 \\ &= \mathrm{E}_{0F}[Y - C_A(t)]^2 - \mathrm{E}_{0F}[\sigma^2(t)] \\ \mathrm{Var}(C) &= \mathrm{E}_F \mathrm{E}_{0F}[C(t, \mathcal{X}) - C_A(t)]^2 \end{aligned} \tag{7}$$

giving the decomposition

$$\mathrm{PE}(Y, C) = \mathrm{E}_{0F}[\sigma^2(t)] + \mathrm{Bias}(C) + \mathrm{Var}(C) \tag{8}$$

Our objective in this article is to construct such a decomposition for classification under general error measures, and to derive bootstrap estimates of its components. Section 2 derives the general decomposition, with the two-class case discussed in Section 3. The bootstrap estimates for bias and variance are given in Section 4, while Section 5 shows an example, featuring linear and nearest-neighbour classifiers. We give the background theory in section 6 and make some final remarks in section 7.

2 A general prediction error decomposition

We assume that $Q[y, c]$ is any error measure that satisfies the conditions given in section 6: this includes the measures in Table 1.

Define the *ideal estimator* by

$$C_0(t) \equiv \mathrm{E}_{0F}(Y|t). \tag{9}$$

$C_0(t)$ is the vector of true class probabilities and $\mathrm{argmax}[C_0(t)]$ is the Bayes rule.

Define the *aggregated predictor* by

$$C_A(t) \equiv \mathrm{E}_F C(t, \mathcal{X}) \tag{10}$$

We imagine drawing an infinite collection of training sets and applying the classifier $C(t, \mathcal{X})$ to each. $C_A(t)$ is the average of $C(t, \mathcal{X})$ at t over this infinite collection. If $C(t, \mathcal{X})$ outputs an indicator vector, then the elements of $C_A(t)$ are the proportions of times each class is predicted in the infinite collection, at input t.

The aggregated predictor $C_A(t)$ reduces the error in $C(t, \mathcal{X})$ by averaging it over training sets drawn from F: we show in section 6 that $C_A(t)$ has smaller expected loss than $C(t, \mathcal{X})$ for any of the loss functions Q in Table 1.

Notice that the classifier $C_A(t)$ may differ depending on whether $C(t, \mathcal{X})$ outputs zeroes and ones, or probabilities. Suppose that at a point t, $C(t, \mathcal{X})$ outputs (.45, .55) or (.9, .1), with probabilities 2/3 and 1/3. Then $C_A(t) = (.6, .4)$ and so predicts the first class. But if $C(t, \mathcal{X})$ outputs the class indicator, that is $(0, 1)$ or $(1, 0)$ with probabilities 2/3 and 1/3, then $C_A(t) = (1/3, 2/3)$ and so predicts the second class.

Breiman (1996) coined the term "aggregated", and called its bootstrap estimate $\hat{C}_A(t) = \mathrm{E}_{\hat{F}}[C(t, \mathcal{X})]$ the bagged ("bootstrap aggregated") predictor. It is called a "bootstrap smoothed" estimate in Efron & Tibshirani (1995). Bagging mimics aggregation by averaging $C(t, \mathcal{X})$ over training sets drawn from $\hat{F}$. In Breiman (1996), bagging is seen to reduce classification error by about 20% on average over a collection of problems. Breiman also reports very little difference when bagging was applied to the class probability estimates, or the corresponding indicator vector for the maximum probability. We discuss bagging further in section 4.

We now define

$$\begin{aligned} \mathrm{Bias}(C) &\equiv \mathrm{PE}(C_0, C_A) \\ \mathrm{Var}(C) &\equiv \mathrm{PE}(C, C_A) \end{aligned} \tag{11}$$

Note that

1. the bias is really a kind of squared bias, as it is always non-negative.
2. C is unbiased if its aggregated version C_A predicts the same class as the Bayes rule, with probability one over the inputs t.
3. the variance is always non-negative
4. if the classifier C does not depend on the training set, then $C = C_A$ and hence its variance is zero.

We could just as well have defined $\mathrm{Var}(C)$ to be $\mathrm{PE}(C_A, C)$. Now as shown in section 6,

$$C_A(t) = \mathrm{argmin}_{C'} \mathrm{PE}(C, C'), \tag{12}$$

so in this sense the definition of variance in (11) is more natural. Of course for squared error the two definitions coincide.

The following result shows that PE satisfies a Pythagorean-type equality.

Lemma 1: For the error measures in Table 1 and others satisfying the conditions given in section 6,

$$\begin{aligned} \text{Bias}(C) &= \text{PE}(C_0, C_A) \\ &= \text{PE}(Y, C_A) - \text{PE}(Y, C_0) \end{aligned} \tag{13}$$

Hence the bias of C is the excess in prediction error of the aggregated predictor C_A over the ideal predictor C_0. The proof is given in section 6. We note however that

$$\text{Var}(C) = \text{PE}(C, C_A) \neq \text{PE}(Y, C) - \text{PE}(Y, C_A) \tag{14}$$

in general. For example, $\text{PE}(C_A, C)$ is always non-negative while $\text{PE}(Y, C) - \text{PE}(Y, C_A)$ need not be. Note that for squared error loss (14) is an equality. Now if the loss function $Q[y, c]$ is convex, then

$$\text{PE}(Y, C_A) \leq \text{PE}(Y, C) \tag{15}$$

by Jensen's inequality. An example is the multinomial deviance loss function $Q_3[y, c]$: for this case (15) holds although the inequality (14) is not an equality.

For non-convex loss functions like misclassification error, (15) need not hold. Here is a simple example. Suppose $Y = 1$ for all t, and the classifier C predicts $Y = 1$ (for all t) with probability .4 and predicts $Y = 0$ (for all t) with probability .6. Then $\text{PE}(Y, C) = .6, \text{PE}(Y, C_A) = 1.0$. As stated by Breiman (1996), aggregation can make a good classifier better but can make a poor classifier worse.

We define the *aggregation effect* as

$$\text{AE}(C) \equiv \text{PE}(Y, C) - \text{PE}(Y, C_A) \tag{16}$$

For squared error loss, $\text{AE}(C) = \text{Var}(C)$, but this does not hold in general for other loss functions. We can think of $\text{AE}(C)$ as being the sum of two terms:

$$\text{AE}(C) = \text{Var}(C) + [\text{PE}(Y, C) - \text{PE}(Y, C_A) - \text{Var}(C)] \tag{17}$$

This can be thought of as the variance plus a term resulting from the shape of the loss function.

It is the aggregation effect, not the variance, that figures directly into prediction error. In particular, we have the decomposition

$$\mathrm{PE}(Y, C) = \mathrm{PE}(Y, C_0) + \mathrm{Bias}(C) + \mathrm{AE}(C) \tag{18}$$

3 The two-class case

The definitions of bias and variance can be written out explicitly in the case of two classes. We consider the misclassification loss Q_1 and squared error loss Q_2.

Let $C(t, \mathcal{X}) = (1 - \hat{p}(t), \hat{p}(t))$, the predictions for each class from the classifier C at input t. The values $\hat{p}(t)$ and $1 - \hat{p}(t)$ will be one or zero, if C predicts a class, but can be between 0 and 1, if C outputs probabilities. Let the true probabilities of class 1 and 2 at t be $C_0 = (1 - p(t), p(t))$.

For simplicity we give the expressions for bias and variance at each fixed input t. The total bias and variance average these expressions test points $(t, y) \sim F$.

The aggregate classifier is $C_A(t) = \mathrm{E}_F(1 - \hat{p}(t), \hat{p}(t)) \equiv (1 - \bar{p}(t), \bar{p}(t))$. For squared error loss,

$$\begin{aligned} \mathrm{Bias}(C)[t] &= \mathrm{E}_F Q_2[C_0(t), C_A(t)] \\ &= 2(\bar{p}(t) - p(t))^2 \\ \mathrm{Var}(C)[t] &= \mathrm{E}_F Q_2[C(t, \mathcal{X}), C_A(t)] \\ &= 2\mathrm{E}_F(\hat{p}(t) - \bar{p}(t))^2 \end{aligned} \tag{19}$$

These are just (twice) the usual bias and variance expressions for $\hat{p}(t)$ as an estimator of $p(t)$.

Now for misclassification loss,

$$\begin{aligned} \mathrm{Bias}(C)[t] &= \mathrm{E}_F Q_1[C_0(t), C_A(t)] \\ &= |2 \cdot p(t) - 1| I(p(t) \geq .5 \;\&\; \bar{p}(t) < .5 \text{ or } p(t) < .5 \;\&\; \bar{p}(t) \geq .5) \\ \mathrm{Var}(C)[t] &= \mathrm{E}_F Q_1[C(t, \mathcal{X}), C_A(t)] \\ &= \mathrm{E}_F |2 \cdot \hat{p}(t) - 1| I(\hat{p}(t) \geq .5 \;\&\; \bar{p}(t) < .5 \text{ or } \hat{p}(t) < .5 \;\&\; \bar{p}(t) \geq .5) \end{aligned} \tag{20}$$

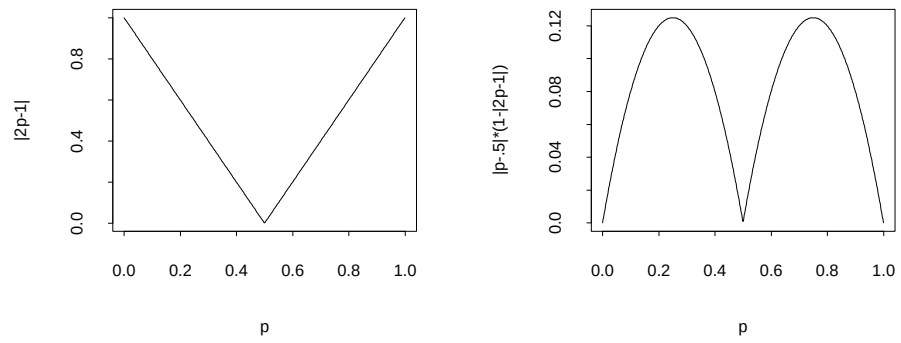

Figure 1: *Left: the function* $|2p-1|$*, part of the bias and variance expressions under misclassification loss. Right: the variance* $|\bar{p} - .5| \cdot (1 - |2\bar{p} - 1|)$ *for 0-1 classification rules under misclassification loss*

Both bias and variance involve the function $|2p - 1|$, shown below in the left panel of Figure 1.

The bias comes from points where $\bar{p}(t)$ and $p(t)$ are on opposite sides of $1/2$, while the variance comes from points where $\hat{p}(t)$ and $\bar{p}(t)$ are on opposite sides of $1/2$. The bias is largest when $p(t)$ is farthest away from $1/2$, that is, near 0 or 1; similarly, the variance depends on how far $\hat{p}(t)$ is from $1/2$.

When the $\hat{p}(t)$ is restricted to be 0 or 1 (that is, the classifier C outputs either $(1, 0)$ or $(0, 1)$), then $\bar{p}(t)$ is the probability that the second class is selected at t, and

$$\mathrm{Var}(C)[t] = |\bar{p}(t) - .5| \cdot (1 - |2\bar{p}(t) - 1|) \tag{21}$$

This function has the interesting shape shown in the right panel of Figure 1, taking its maxima at $p = 1/4$ and $p = 3/4$. Note that if the classification rule always predicts either 0 or 1 at a point t, then $\bar{p}(t)$ is either 0 or 1, and hence $\mathrm{Var}(C)[t] = 0$.

4 Bootstrap estimates of bias and variance

Using the bootstrap method we can derive a sample-based estimate of the prediction error decomposition (18). The generic bootstrap approach plug-ins the empirical distribution $\hat{F}$ for F. However here we have two distributions to estimate, the training sample distribution F and the test sample distribution $0F$. As argued in Efron & Tibshirani (1995), use of $\hat{F}$ for both estimates leads to a large downward bias since the support of the training and test samples overlap. To avoid this, we use the leave one-out bootstrap approach of Efron & Tibshirani (1995). We first give details of the bootstrap estimate of the variance component. From (13) the variance of the classifier C is

$$\mathrm{Var}(C) = \mathrm{E}_{0F}\mathrm{E}_F Q[C(t,\mathcal{X}), C_A(t,F)] \tag{22}$$

where we have made explicit the dependence of C_A on F.

Let $\hat{F}_{(i)}$ be the distribution putting mass $1/(N-1)$ on all of the points except x_i, where it puts zero mass. Then our estimate is

$$\begin{aligned} \widehat{\mathrm{Var}}(C) &= \mathrm{E}_{\hat{F}}\mathrm{E}_{\hat{F}_{(i)}} Q[C(t,\mathcal{X}^*), C_A(t,\hat{F}_{(i)})] \\ &= \frac{1}{N}\sum \mathrm{E}_{\hat{F}_{(i)}} Q[C(t_i,\mathcal{X}^*), C_A(t_i,\hat{F}_{(i)}),] \end{aligned} \tag{23}$$

where $\mathcal{X}^*$ is a bootstrap training set drawn from $\hat{F}_{(i)}$. We can estimate $\widehat{\mathrm{Var}}(C)$ from a single set of Monte-Carlo samples:

1. Draw bootstrap training sets $\mathcal{X}^{1*}, \mathcal{X}^{2*}, \ldots \mathcal{X}^{*B}$ with replacement from the training set $\mathcal{X}$. Compute the classifier $C(t,\mathcal{X}^{*b})$ on each.

2. Let V_i be the indices of the bootstrap training sets that do not contain observation i. For each i, construct the aggregate classifier $C_A(t,\hat{F}_{(i)})$ from the classifiers $C(t,\mathcal{X}^{*b})$ for $b \in V_i$:

$$\hat{C}_{A,\mathrm{Boot}}(t,\hat{F}_{(i)}) = \sum_{b\in V_i} C(t_i,\mathcal{X}^{*b})/B_i, \tag{24}$$

   where B_i is the number of bootstrap samples in V_i. Our estimate of $\mathrm{Var}(C)$ is given by

$$\widehat{\mathrm{Var}}_{\mathrm{Boot}}(C) = \frac{1}{N}\sum_{i=1}^{N}\sum_{b\in V_i} Q[C(t_i,\mathcal{X}^{*b}), \hat{C}_A(t,\hat{F}_{(i)})]/B_i, \tag{25}$$

The bootstrap estimate of the aggregated predictor $C_A(t) = \mathrm{E}_F[C(t, \mathcal{X})]$ is

$$\hat{C}_A(t) = \mathrm{E}_{\hat{F}}[C(t, \mathcal{X}^*)] \tag{26}$$

This is the "bagged" estimate of Breiman (1996) and the "bootstrap smoothed" estimate of Efron & Tibshirani (1995) The leave-one-out bootstrap estimate of error for $\hat{C}_A$ is

$$\widehat{\mathrm{Err}}^{(1)}(\hat{C}_A) = \frac{1}{N} \sum \mathrm{E}_{\hat{F}_{(i)}} Q[y_i, C_A(t_i, \hat{F}_{(i)})] \tag{27}$$

This is estimated from Monte Carlo samples in the same manner as the variance above, namely

$$\widehat{\mathrm{Err}}^{(1)}_{\mathrm{Boot}}(\hat{C}_A) = \frac{1}{N} \sum_{i=1}^{N} \sum_{b \in V_i} Q[Y_i, \hat{C}_A(t, \hat{F}_{(i)})]/B_i, \tag{28}$$

Let

$$\widehat{\mathrm{Err}}^{(1)}(C) = \frac{1}{N} \sum \mathrm{E}_{\hat{F}_{(i)}} Q[y_i, C(t_i, \hat{F}_{(i)})], \tag{29}$$

where $\hat{F}_{(i)}$ is the distribution putting mass $1/(N-1)$ on each of the training points except the ith one. This is called the leave-one-out bootstrap estimate of error for C. This is estimated from Monte Carlo samples in the same manner as above. Thus we estimate the aggregation effect $\mathrm{AE}(C)$ by

$$\widehat{\mathrm{AE}}(C) = [\widehat{\mathrm{Err}}^{(1)}(C) - \widehat{\mathrm{Err}}^{(1)}(\hat{C}_A)] \tag{30}$$

Estimation of the bias term is more problematic. Using (13), and letting $\mathrm{PE}_{\mathrm{Bayes}} = \mathrm{PE}(Y, C_0)$ we can form the estimate

$$\begin{aligned} \widehat{\mathrm{Bias}}(C) &= \widehat{\mathrm{Err}}^{(1)}(\hat{C}_A) - \mathrm{PE}_{\mathrm{Bayes}} \\ &\leq \widehat{\mathrm{Err}}^{(1)}(\hat{C}_A) \end{aligned} \tag{31}$$

The quantity $\widehat{\mathrm{Err}}^{(1)}(\hat{C}_A)$ provides an approximate upper bound for $\mathrm{Bias}(C)$.

A better bound might be obtained by getting an estimate of the Bayes risk $\mathrm{PE}_{\mathrm{Bayes}}$, but this is difficult to estimate well. Some methods for this

have been suggested in the literature: for example, if PE_{NN} is the prediction error of the 1- nearest neighbour classifier, then Cover & Hart (1967) give the (asymptotic) upper bound $\mathrm{PE}_{NN} \leq \mathrm{PE}_{\mathrm{Bayes}}(1 - \mathrm{PE}_{\mathrm{Bayes}}/\alpha)$ where $\alpha = (K-1)/K$. This gives the asymptotic lower bound

$$\mathrm{PE}_{\mathrm{Bayes}} \geq \alpha - [\alpha(\alpha - \mathrm{PE}_{NN})]^{1/2} \tag{32}$$

Using this we can obtain the approximate upper bound on the bias of the classifier C:

$$\mathrm{Bias}(C) \leq \widehat{\mathrm{Err}}^{(1)}(\hat{C}_A) - \{\alpha - [\alpha(\alpha - \mathrm{PE}_{NN})]^{1/2}\} \tag{33}$$

Putting together the various components, we have a sample based decomposition corresponding to (18). This has the form

$$\widehat{\mathrm{Err}}^{(1)}(C) = \mathrm{PE}_{\mathrm{Bayes}} + \left(\widehat{\mathrm{Err}}^{(1)}(C_A) - \mathrm{PE}_{\mathrm{Bayes}}\right) + \widehat{\mathrm{AE}}(C) \tag{34}$$

5 An Example

We illustrate these estimates, using misclassification loss Q_1, on two problems:

Two Normals- centered at (0,0,0,0) and (2,0,0,0) in 4 dimensions, with identity covariance matrices and $N = 50$ cases in each training set.

Concentric normals- the first class is centered at (0,0,0,0) with identity covariance. The second has the same distribution as the first, but conditioned on the squared length of the x vector being between 9 and 16. Thus the second class almost completely surrounds the first. There are $N = 50$ cases in each training set.

We ran a linear discriminant and 5-nearest neighbour classifier on ten samples from each of these models, with each classifier outputting an indicator vector for the predicted class. The results are shown in Table 2. The "True values" shown are computed using $\hat{F}$ to generate training sample and the true F to generate a large test sample.

The leave-one out estimate of variance does a reasonable job of approximating the true variance, while the bias upper bound is sometimes far too

Table 2: *Results for 2-norm and concentric normal problems. Values are mean (standard deviation) over 10 runs. LDF is linear discriminant function, 5-NN is 5 nearest neighbour classifier. The estimated* Bias(C) *values are approximate upper bounds from (33).*

		Two normals		Concentric normals	
	Bayes rate	.159		.058	
		True value	Estimate	True value	Estimate
LDF					
	PE(Y, C)	0.176(0.006)	0.191(0.011)	0.488(0.007)	0.506(0.021)
	Var(C)	0.114(0.065)	0.080(0.004)	0.278(0.049)	0.293(0.023)
	PE($\hat{C}_A, C$)	0.031(0.003)	0.034(0.002)	0.096(0.007)	0.104(0.007)
	Bias(C)	0.018(0.007)	0.019(0.007)	0.422(0.008)	0.319(0.030)
	PE($Y, \hat{C}_A$)	0.177(0.007)	0.166(0.011)	0.480(0.008)	0.490(0.028)
	AE(C)	-0.001(0.002)	0.025(0.007)	0.007(0.004)	0.016(0.009)
5-NN					
	PE(Y, C)	0.250(0.010)	0.234(0.012)	0.254(0.005)	0.342(0.015)
	Var(C)	0.164(0.085)	0.130(0.007)	0.298(0.045)	0.207(0.009)
	PE($\hat{C}_A, C$)	0.102(0.004)	0.058(0.003)	0.150(0.012)	0.083(0.004)
	Bias(C)	0.039(0.007)	0.051(0.016)	0.278(0.005)	0.159(0.013)
	PE($Y, \hat{C}_A$)	0.198(0.007)	0.198(0.018)	0.336(0.005)	0.328(0.020)
	AE(C)	0.052(0.008)	0.036(0.008)	-0.081(0.006)	0.014(0.010)

low. In the case of 5-NN for the concentric normal problem, the bagged estimate has much larger prediction error that the estimate itself. The reason is that bagging nearest neighbours has the effect of increasing the number of neighbours used. In this problem, the fewer neighbours used, the better.

6 Theoretical background

In this section we provide the background theory for Lemma 1, and the prediction error decomposition.

We first extend some of the binary data definitions of Efron (1978) to the general K-class case. As before, let e_k be the kth unit vector of length K and suppose $y = (y^1, \ldots y^K) \in \{e_1, e_2, \ldots e_K\}$. If $\pi = (\pi^1, \ldots \pi^K)$ is a vector of probabilities adding to 1, we begin with a variation function $Q[y, \pi]$ between y and π. Misclassification error is defined by the particular choice

$$Q_1[y, \pi] = \begin{cases} 1 & \text{if } \text{argmax}(y) \notin \text{argmax}(\pi) \\ \frac{m-1}{m} & \text{if } \text{argmax}(y) \in \text{argmax}(\pi),\ m = \#\text{argmax}(\pi) \end{cases} \tag{35}$$

If π has a single largest element, $Q_1[y, \pi]$ just counts a classification error. In the case of a tie, it is the probability of an error, assuming we pick at random among the maximum probability classes in π.

Another common choice for Q is squared error

$$Q_2[y, \pi] = \sum_k (y^k - \pi^k)^2 \tag{36}$$

The general requirements for $Q[\cdot, \cdot]$ are as follows. If P permutes the elements of a K-vector, we require for all k:

$$\begin{aligned} Q[Pe_k, P\pi] &= Q[e_k, \pi] \\ Q[e_k, e_k] &= 0 \end{aligned} \tag{37}$$

We also that $Q[e_k, \pi]$ be non-increasing in $||e_k - \pi||^2$. This ensures that however Q is measuring loss, it doesn't decrease as the probability vector π gets farther away from e_k.

Let $s_k(\pi) = Q[e_k, \pi]$, and define the dispersion function for a vector $p = (p^1, \ldots p^K)$ to be :

$$\sigma(p) = \sum_k p^k s_k(p) \tag{38}$$

This function measures the internal dispersion of the probability vector p. For misclassification error $\sigma(p) = 1 - \max(p)$; for squared error $\sigma(p) = \sum_{j \neq k} p^j p^k$. The last requirement for $Q[\cdot, \cdot]$ is that

$$\sigma(p) = \min_{\pi} \{ \sum_k p^k s_k(\pi) \} \tag{39}$$

so that $\sigma(\cdot)$ is concave.

Through a geometric argument of Efron (1978), we extend the definition of $Q[\cdot, \cdot]$ for cases where the first argument is any probability vector (not just one of the e_k). We define

$$Q[p, \pi] = \sum p^k s_k(\pi) - \sigma(p) \tag{40}$$

For misclassification error

$$Q_1[p, \pi] = p^a - \sum_{b \in B} p^b / |B| \tag{41}$$

where $a = \text{argmax}(p), B = \text{argmax}(\pi), |B|$ equals the number of elements in B; for squared error,

$$Q_2[p, \pi] = \sum_k (p^k - \pi^k)^2 \tag{42}$$

Note that (39) and (40) imply

$$\begin{aligned} C_0(t) &= \text{E}(Y|t) = \text{argmin}_C \text{PE}(Y, C) \\ C_A(t) &= \text{E}(C|t) = \text{argmin}_{C'} \text{PE}(C, C') \end{aligned} \tag{43}$$

As an aside, let $\mathbf{Y}$ be a $n \times K$ matrix of observations, having rows Y_i and $\mathbf{P}$ a corresponding matrix of probabilities with rows P_i. Define

$$\mathbf{Q}[\mathbf{Y}, \mathbf{P}] = \sum Q[Y_i, P_i] \tag{44}$$

The function $\mathbf{Q}$ satisfies Pythagorean-type relations. Consider for example a one-way layout. Let $\hat{\mathbf{P}}$ be the matrix of within group proportions, and $\tilde{\mathbf{P}}$ the matrix of observed overall proportions, and $\mathbf{\Pi}$ be an arbitrary matrix of overall proportions. Then it can shown that

$$\mathbf{Q}[\mathbf{Y}, \tilde{\mathbf{P}}] = \mathbf{Q}[\mathbf{Y}, \hat{\mathbf{P}}] + \mathbf{Q}[\hat{\mathbf{P}}, \tilde{\mathbf{P}}]$$

$$\begin{aligned} \mathbf{Q}[\mathbf{Y},\mathbf{\Pi}] &= \mathbf{Q}[\mathbf{Y},\hat{\mathbf{P}}] + \mathbf{Q}[\mathbf{\Pi},\tilde{\mathbf{P}}] \\ \mathbf{Q}[\mathbf{Y},\mathbf{\Pi}] &= \mathbf{Q}[\mathbf{Y},\hat{\mathbf{P}}] + \mathbf{Q}[\hat{\mathbf{P}},\tilde{\mathbf{P}}] + \mathbf{Q}[\tilde{\mathbf{P}},\mathbf{\Pi}] \end{aligned} \tag{45}$$

Now Lemma 1 says that $Q[\cdot,\cdot]$ satisfies a similar relation at the population level.

Proof of Lemma 1:

$$\begin{aligned} \mathrm{PE}(Y,C_A) - PE(Y,C_0) &= \mathrm{E}\{\sum Y^k[s_k(C_A) - s_k(Y)] - \sum Y^k[s_k(C_0) - s_k(Y)]\} \\ &= \mathrm{E}\{\sum Y^k[s_k(C_A) - s_k(C_0)]\} \\ &= \mathrm{E}\{\sum C_0^k[s_k(C_A) - s_k(C_0)]\} \\ &= \mathrm{PE}(C_0,C_A) \end{aligned} \tag{46}$$

since $\mathrm{E}(Y|t) = C_0(t)$.

7 Discussion

In this paper we have discussed a general decomposition of prediction error for classifiers, and derived bootstrap estimates of the various components. From a practical viewpoint, the bootstrap error estimate for the bagged classifier is useful for determining whether bagging is useful for a given problem.

Recently, other authors have studied the question of bias and variance in classification. Kohavi & Wolpert (1996) give a decomposition for misclassification error that is a special case of our decomposition (18), that results from the use of the squared error loss function $Q_2[y,c]$. Although $Q_1[y,c] = .5 \cdot Q_2[y,c]$ when y and c are 0-1 vectors (one of $e_1,\ldots e_K$), they are not the same in general. As a result the definitions of bias and variance are materially different. For example in a two class problem if the true probability of class 2 is 0.9 (for all t) and the aggregate classifier C_A predicts class 2 with probability 0.6 (for all t), then $\mathrm{Bias}(C) = 0$ under misclassification error, but $\mathrm{Bias}(C) = 0.3^2$ under squared error. From a misclassification error viewpoint, the classifier C_A is the good as the Bayes rule. When viewed as an estimate of the class probabilities, it is not.

Breiman (1996) approaches the problem as follows. Let U be the set of values of t where the classifier C is unbiased, that is, $C_A(t) = C_0(t)$. Let B be the bias set, the complement of U. Then Breiman defines

$$\mathrm{Bias}(C) = P(C_0(t) = Y, t \in B) - \mathrm{E}[P(C(t,\mathcal{X}) = Y, t \in B)]$$

$$\text{Var}(C) \quad = \quad P(C_0(t) = Y, t \in U) - \text{E}[P(C(t, \mathcal{X}) = Y, t \in U)] \qquad (47)$$

The probabilities in each expression average over t and Y. The expectations average over training sets $\mathcal{X}$. These definitions lead to an exact additive decomposition of prediction error into Bayes error, bias and variance. However they seem to be artificially constructed to achieve an additive decomposition of prediction error. For example, the set U may be empty, that is, $C_A(t)$ doesn't agree with $C_0(t)$ for any t. Then the variance would be either undefined or zero, neither of which is satisfactory.

The definitions of bias and variance introduced in the present paper are natural in that they are expressed directly in terms of the loss function. The non-additivity of prediction error that results when the loss function is misclassification error results from the non-convexity, and seems to be a fundamental aspect of the problem.

Friedman (1996) looks at the two class problem, decomposing misclassification error into the bias and variance of the estimated probabilities (as opposed to the bias and variance of the classification rule, as is done here). He shows that bias and variance do not add, but can interact in an interesting way. This can have important consequences for selection of tuning parameters for classifiers. For example, Friedman illustrates how the neighborhood size for a K-nearest neighbour classifier should be chosen much larger under misclassification loss and than squared error loss.

The decomposition and bootstrap estimates can be applied to other problems, such as regression under squared error loss and generalized regression in the exponential family. This is an interesting topic for further research.

Acknowledgments

I would like to thank Bradley Efron, Trevor Hastie, a referee and an associate editor for helpful discussions. Support from the Natural Sciences and Engineering Research Council of Canada and the IRIS Centres of Excellence is gratefully acknowledged.

References

Breiman, L. (1996), Bias, variance and arcing classifiers, Technical report, University of California, Berkeley.

Cover, T. & Hart, P. (1967), 'Nearest neighbor pattern classification', *Proc. IEEE Trans. Inform. Theory* pp. 21–27.

Efron, B. (1978), 'Regression and anova with zero-one data', *J. Amer. Statist. Assoc.* pp. 113–121.

Efron, B. & Tibshirani, R. (1995), Cross-validation and the bootstrap: Estimating the error rate of a prediction rule, Technical report, Stanford University.

Friedman, J. (1996), Bias, variance, 0-1 loss and the curse of dimensionality, Technical report, Stanford University.

Kohavi, R. & Wolpert, D. H. (1996), Bias plus variance decomposition for zero-one loss functions, *in* L. Saitta, ed., 'Machine Learning: Proceedings of the Thirteenth International Conference', Morgan Kaufmann Publishers, Inc. Available at http://robotics.stanford.edu/users/ronnyk.