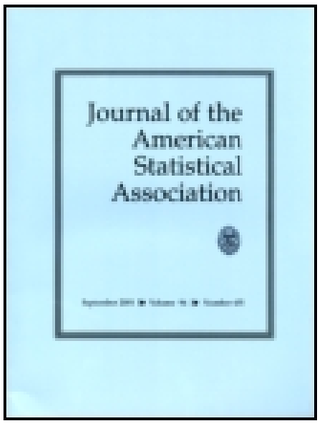

Journal of the American Statistical Association
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uasa20

Combining Estimates in Regression and Classification

Michael Leblanc [a] & Robert Tibshirani [b]
[a] Fred Hutchinson Cancer Research Center , Seattle , WA , 98104
[b] Department of Preventive Medicine and Biostatistics and Department of Statistics , University of Toronto , Ontario , M5S 1A8 , Canada
Published online: 27 Feb 2012.

To cite this article: Michael Leblanc & Robert Tibshirani (1996) Combining Estimates in Regression and Classification, Journal of the American Statistical Association, 91:436, 1641-1650, DOI: 10.1080/01621459.1996.10476733

To link to this article: http://dx.doi.org/10.1080/01621459.1996.10476733

Combining Estimates in Regression and Classification

Michael LEBLANC and Robert TIBSHIRANI

We consider the problem of how to combine a collection of general regression fit vectors to obtain a better predictive model. The individual fits may be from subset linear regression, ridge regression, or something more complex like a neural network. We develop a general framework for this problem and examine a cross-validation–based proposal called "model mix" or "stacking" in this context. We also derive combination methods based on the bootstrap and analytic methods and compare them in examples. Finally, we apply these ideas to classification problems where the estimated combination weights can yield insight into the structure of the problem.

KEY WORDS: Bootstrap; Cross-validation; Model combination.

1. INTRODUCTION

Consider a standard regression setup: predictor measurements $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$ and a response measurement y_i on N independent training cases. Let $\mathbf{z}$ represent the entire training sample. Our goal is derive a function $c_{\mathbf{z}}(\mathbf{x})$ that accurately predicts future y values.

Suppose that we have available K different regression fitted values from these data, denoted by $c_{\mathbf{z}}^k(\mathbf{x})$, for $k = 1, 2, \ldots, K$. For example, $c_{\mathbf{z}}^k(\mathbf{x})$ might be a least squares fit for some subset of the variables, or a ridge regression fit, or something more complicated like the result of a projection-pursuit regression or neural network model. Or the collection of $c_{\mathbf{z}}^k(\mathbf{x})$'s might correspond to a single procedure run with K different values of an adjustable parameter. In this article we consider the problem of how to best combine these estimates to obtain an estimator that is better than any of the individual fits. The class that we consider has the form of a simple linear combination:

$$\sum_{k=1}^{K} \beta_k c_{\mathbf{z}}^k(\mathbf{x}). \tag{1}$$

One way to obtain estimates of $\beta_1, \ldots, \beta_K$ is by least squares regression of y on $c_{\mathbf{z}}^1(\mathbf{x}), \ldots, c_{\mathbf{z}}^K(\mathbf{x})$. But this might produce poor estimates, because it does not take into account the relative amount of fitting present in each of the $c_{\mathbf{z}}^k(\mathbf{x})$'s, or the correlation of the $c_{\mathbf{z}}^k(\mathbf{x})$'s induced by the fact that all of the regression fits are estimated from the data $\mathbf{z}$. For example, if the $c_{\mathbf{z}}^k(\mathbf{x})$'s represent a nested set of linear models, then β_k will equal 1 for the largest model and zero for the others, and hence will simply reproduce the largest model.

In a paper in the neural network literature, Wolpert (1992) presented an interesting idea known as "stacked generalization" for combining estimators. His proposal was translated into statistical language by Breiman (1995), who applied and studied it the regression setting, calling it "stacked regression." Wolpert and Breiman were apparently unaware that stacked regression is exactly the same as the "model-mix" proposal of Stone (1974, ex. 3.5)—a fact that we just recently learned.

Here is how the model-mix idea works. We let $c_{\mathbf{z}^{(-i)}}^k(\mathbf{x}_i)$ denote the leave-one-out cross-validated fit for $c_{\mathbf{z}}^k(\mathbf{x})$, evaluated at $\mathbf{x} = \mathbf{x}_i$. The model-mix method minimizes

$$\sum_{i=1}^{N} \left[y_i - \sum_{k=1}^{K} \beta_k c_{\mathbf{z}^{(-i)}}^k(\mathbf{x}_i) \right]^2, \tag{2}$$

producing estimates $\hat{\beta}_1, \ldots, \hat{\beta}_K$. The final predictor function is $v_{\mathbf{z}}(\mathbf{x}) = \sum \hat{\beta}_k c_{\mathbf{z}}^k(\mathbf{x})$.

Notice how this differs from a more standard use of cross-validation. Usually, for each method k, one constructs the prediction error estimate

$$\widehat{\mathrm{PE}}(k) = \frac{1}{N} \sum_{i=1}^{N} [y_i - c_{\mathbf{z}^{(-i)}}^k(\mathbf{x}_i)]^2, \tag{3}$$

then chooses the $c_{\mathbf{z}}^k(\mathbf{x})$ that minimizes $\widehat{\mathrm{PE}}(k)$. Here we are estimating a linear combination of models rather than choosing just one.

In the particular cases that he tried, Breiman found that the linear combination $\sum_{k=1}^{K} \hat{\beta}_k c_{\mathbf{z}}^k(\mathbf{x})$ did not exhibit good prediction performance. However, when the coefficients in (2) were constrained to be nonnegative, $v_{\mathbf{z}}(\mathbf{x})$ showed better prediction error than any of the individual $c_{\mathbf{z}}^k(\mathbf{x})$. In some cases the improvement was substantial.

This article grew out of our attempt to understand how and why the model-mix method works. Cross-validation is usually used to estimate the prediction error of an estimator. At first glance, model-mix seems to use cross-validation for a fundamentally different purpose—namely, to construct a new estimator from the data. Here we derive a framework for the problem of combining estimators, and as a result, we cast the model-mix method into more familiar terms. We also derive combination estimators based on the bootstrap, and, in some simple cases, on analytic methods.

Michael LeBlanc is Associate Member, Fred Hutchinson Cancer Research Center, Seattle, WA 98104. Robert Tibshirani is Professor, Department of Preventive Medicine and Biostatistics and Department of Statistics, University of Toronto, Ontario, M5S 1A8, Canada. The authors thank Mike Escobar, Trevor Hastie, Geoffrey Hinton, and David Wolpert for helpful discussions and suggestions. They gratefully acknowledge the support of the Natural Science and Engineering Research Council of Canada. The second author also acknowledges support from the U.S. National Institutes of Health through grant NCI 2 P01 CA 53996.

2. A FRAMEWORK FOR COMBINING REGRESSION ESTIMATORS

As in Section 1, here we have data $(\mathbf{x}_i, y_i), i = 1, 2, \ldots, N$, with $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, and let $\mathbf{z}$ be the entire sample. We assume that each pair $(\mathbf{x}_i, y_i)$ is an independent realization of random variables $(\mathbf{X}, Y)$ having distribution F.

Let $C_{\mathbf{z}}(\mathbf{x}_0) = (c_{\mathbf{z}}^1(\mathbf{x}_0), \ldots, c_{\mathbf{z}}^K(\mathbf{x}_0))^T$ be a K vector of estimators evaluated at $\mathbf{X} = \mathbf{x}_0$, based on data $\mathbf{z}$.

We seek coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)^T$, so that the linear combination estimator $C_{\mathbf{z}}(\mathbf{x})^T\boldsymbol{\beta} = \sum \beta_k c_{\mathbf{z}}^k(\mathbf{x})$ has low prediction error. Our criterion for selecting $\boldsymbol{\beta}$ is

$$\tilde{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} E_{0F}(Y_0 - C_{\mathbf{z}}(\mathbf{X}_0)^T\boldsymbol{\beta})^2. \tag{4}$$

Here $\mathbf{z}$ is fixed and E_{0F} denotes expectation under $(\mathbf{X}_0, Y_0) \sim F$.

Of course, in real problems we do not know E_{0F}, and so we must derive sample-based estimates. The obvious estimate of $g(\mathbf{z}, F, \boldsymbol{\beta}) = E_{0F}(Y_0 - C_{\mathbf{z}}(\mathbf{X}_0)^T\boldsymbol{\beta})^2$ is the plug-in or resubstitution estimate

$$\begin{aligned} g(\mathbf{z}, \hat{F}, \boldsymbol{\beta}) &= E_{0\hat{F}}(Y_0 - C_{\mathbf{z}}(\mathbf{X}_0)^T\boldsymbol{\beta})^2 \\ &= \frac{1}{N}\sum_1^N (y_i - C_{\mathbf{z}}(\mathbf{x}_i)^T\boldsymbol{\beta})^2, \end{aligned} \tag{5}$$

whose minimizer is the least squares estimator

$$\hat{\boldsymbol{\beta}}_{\mathrm{LS}} = \left[\sum_i C_{\mathbf{z}}(\mathbf{x}_i)C_{\mathbf{z}}(\mathbf{x}_i)^T\right]^{-1}\left[\sum_i C_{\mathbf{z}}(\mathbf{x}_i)y_i\right]. \tag{6}$$

What is wrong with $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$? The problem is that the data $\mathbf{z} = (\mathbf{x}_1, y_1,) \cdots (\mathbf{x}_n, y_n)$ are used in two places: in constructing the $C_{\mathbf{z}}$ and in evaluating the error between y_i and $C_{\mathbf{z}}(x_i)$. As a result, the following hold:

a. $\sum_i C_{\mathbf{z}}(\mathbf{x}_i)C_{\mathbf{z}}(\mathbf{x}_i)^T$ and $\sum_i C_{\mathbf{z}}(\mathbf{x}_i)y_i$ are biased estimates of their population analogs.
b. $g(\mathbf{z}, \hat{F}, \boldsymbol{\beta})$ is a biased estimator of $g(\mathbf{z}, F, \boldsymbol{\beta})$.

Although description a may seem to be more direct, in Section 3 we show that description b is more useful.

The model-mix or stacking method estimates the prediction error $g(\mathbf{z}, F, \boldsymbol{\beta})$ by

$$\frac{1}{N}\sum_1^N (y_i - C_{\mathbf{z}^{(-i)}}(\mathbf{x}_i)^T\boldsymbol{\beta})^2. \tag{7}$$

The corresponding minimizer gives the stacking estimator of $\tilde{\boldsymbol{\beta}}$:

$$\hat{\boldsymbol{\beta}}_{\mathrm{St}} = \left[\sum_i C_{\mathbf{z}^{(-i)}}(\mathbf{x}_i)C_{\mathbf{z}^{(-i)}}(\mathbf{x}_i)^T\right]^{-1}\sum_i C_{\mathbf{z}^{(-i)}}(\mathbf{x}_i)y_i. \tag{8}$$

Stacking uses different data to construct C and to evaluate the error between y and C, and hence should produce a less-biased estimator of $g(\mathbf{z}, F, \boldsymbol{\beta})$.

Remark A. Note that there is no intercept in the linear combination in (4). This makes sense when each estimator $c_{\mathbf{z}}^k(\mathbf{x})$ is an estimator of Y, so that $E[c_{\mathbf{z}}^k(\mathbf{x})] \approx EY$, and hence there are values of $\boldsymbol{\beta}$ giving $E[\sum \beta_k c_{\mathbf{z}}^k(\mathbf{x})] \approx EY$ (e.g., take $\sum \beta_k = 1$). In the more general case, each $c_{\mathbf{z}}^k(\mathbf{x})$ does not necessarily estimate Y; for example, each $c_{\mathbf{z}}^k(\mathbf{x})$ might be an adaptively chosen basis function in a nonlinear regression model. Then we should want to include an intercept in the linear combination. This causes no difficulty in the foregoing framework, because we can just set $c_{\mathbf{z}}^1(\mathbf{x}) \equiv 1$.

3. BOOTSTRAP ESTIMATES

One can apply the bootstrap to this problem by bias-correcting the quantities $\sum_i C_{\mathbf{z}}(\mathbf{x}_i)C_{\mathbf{z}}(\mathbf{x}_i)^T$ and $\sum_i C_{\mathbf{z}}(\mathbf{x}_i)y_i$ appearing in the least squares estimator (6). However, it is more useful to approach the problem in terms of bias correction of the prediction error function $g(\mathbf{z}, \hat{F}, \boldsymbol{\beta})$. We then minimize the bias-corrected function $\tilde{g}(\mathbf{z}, \hat{F}, \boldsymbol{\beta})$, to obtain an improved estimator $\tilde{\boldsymbol{\beta}}$. The estimator that we obtain from this procedure is in fact the least squares estimator that uses bias-corrected versions of $\sum_i C_{\mathbf{z}}(\mathbf{x}_i)C_{\mathbf{z}}(\mathbf{x}_i)^T$ and $\sum_i C_{\mathbf{z}}(\mathbf{x}_i)y_i$. The advantage of approaching the problem through the prediction error function is that regularization of the estimator can be incorporated in a straightforward manner (see Sec. 5).

The method presented here is somewhat novel in that we bias-correct a *function* of $\boldsymbol{\beta}$ rather than a single estimator, as is usually the case. A similar proposal in a different setting has been presented by McCullagh and Tibshirani (1988).

The bias of $g(\mathbf{z}, \hat{F}, \boldsymbol{\beta})$ is

$$\Delta(F, \boldsymbol{\beta}) = E_F[g(\mathbf{z}, F, \boldsymbol{\beta}) - g(\mathbf{z}, \hat{F}, \boldsymbol{\beta})]. \tag{9}$$

Given an estimate $\hat{\Delta}(F, \boldsymbol{\beta})$, our improved estimate is

$$\hat{g}(\mathbf{z}, F, \boldsymbol{\beta}) = g(\mathbf{z}, \hat{F}, \boldsymbol{\beta}) + \hat{\Delta}(F, \boldsymbol{\beta}). \tag{10}$$

Finally, an estimate of $\boldsymbol{\beta}$ is obtained by minimization of expression (10).

How can we estimate $\Delta(F, \boldsymbol{\beta})$? Note that the model-mix method implicitly uses the quantity

$$\frac{1}{N}\sum_1^n (y_i - C_{\mathbf{z}^{(-i)}}(\mathbf{x}_i)^T\boldsymbol{\beta})^2 - \frac{1}{N}\sum_1^n (y_i - C_{\mathbf{z}}(\mathbf{x}_i)^T\boldsymbol{\beta})^2$$

to estimate $\Delta(F, \boldsymbol{\beta})$.

Here we outline a bootstrap method that is similar to the estimation of the optimism of an error rate (Efron 1982; Efron and Tibshirani 1993, chap. 17). The bootstrap estimates $\Delta(F, \boldsymbol{\beta})$ by

$$\Delta(\hat{F}, \boldsymbol{\beta}) = E_{\hat{F}}[g(\mathbf{z}^*, \hat{F}, \boldsymbol{\beta}) - g(\mathbf{z}^*, \hat{F}^*, \boldsymbol{\beta})], \tag{11}$$

where $\hat{F}^*$ is the empirical distribution of a sample drawn with replacement from $\mathbf{z}$.

The advantage of the simple linear combination estimator is that the resulting minimizer of (10) can be written down explicitly (see Sec. 8). Let

$$g_1(\mathbf{z}, \hat{F}) = \frac{1}{N}\sum_1^N C_{\mathbf{z}}(\mathbf{x}_i)C_{\mathbf{z}}(\mathbf{x}_i)^T,$$

$$g_2(\mathbf{z}, \hat{F}) = \frac{1}{N}\sum_1^N C_{\mathbf{z}}(\mathbf{x}_i)y_i,$$

$$\Delta_1(\hat{F}) = E_{\hat{F}}[g_1(\mathbf{z}^*, \hat{F}) - g_1(\mathbf{z}^*, \hat{F}^*)],$$

and

$$\Delta_2(\hat{F}) = E_{\hat{F}}[g_2(\mathbf{z}^*, \hat{F}) - g_2(\mathbf{z}^*, \hat{F}^*)]. \quad (12)$$

Then the minimizer of (10) is

$$\hat{\boldsymbol{\beta}}_{\mathrm{Bo}} = [g_1(\mathbf{z}, \hat{F}) + \Delta_1(\hat{F})]^{-1}[g_2(\mathbf{z}, \hat{F}) + \Delta_2(\hat{F})]. \quad (13)$$

In simple terms, we bias-correct both $\sum_1^N C_{\mathbf{z}}(\mathbf{x}_i)C_{\mathbf{z}}(\mathbf{x}_i)^T/N$ and $\sum_1^N C_{\mathbf{z}}(\mathbf{x}_i)y_i/N$, and the resulting estimator is just the least squares estimator that uses the bias-corrected versions.

As with most applications of the bootstrap, we must use bootstrap sampling (Monte Carlo simulation) to approximate these quantities. Full details are given in the following algorithm:

1. Draw B bootstrap samples $\mathbf{z}^{*b} = (\mathbf{x}^{*b}, y^{*b}), b = 1, 2, \ldots, B$, and from each sample derive the estimators $C_{\mathbf{z}^{*b}}$.

2. Evaluate $C_{\mathbf{z}^{*b}}$ on both the original sample and on the bootstrap sample, and compute

$$\hat{\Delta}_1 = \frac{1}{B}\sum_{b=1}^B \left[\frac{1}{N}\sum_{i=1}^N C_{\mathbf{z}^{*b}}(x_i)C_{\mathbf{z}^{*b}}(x_i)^T - \frac{1}{N}\sum_{i=1}^N C_{\mathbf{z}^{*b}}(x_i^{*b})C_{\mathbf{z}^{*b}}(x_i^{*b})^T\right]$$

and

$$\hat{\Delta}_2 = \frac{1}{B}\sum_{b=1}^B \left[\frac{1}{N}\sum_{i=1}^N C_{\mathbf{z}^{*b}}(x_i)y_i - \frac{1}{N}\sum_{i=1}^N C_{\mathbf{z}^{*b}}(x_i^{*b})y_i^{*b}\right].$$

This gives corrected variance and covariances

$$M_{CC} = \frac{1}{N}\sum_{i=1}^N C_{\mathbf{z}}(x_i)C_{\mathbf{z}}(x_i)^T + \hat{\Delta}_1$$

and

$$M_{Cy} = \frac{1}{N}\sum_{i=1}^N C_{\mathbf{z}}(x_i)y_i + \hat{\Delta}_2.$$

3. Use M_{CC} and M_{Cy} to produce a (possibly regularized) regression of y on C. The regression coefficients $\hat{\beta}$ are the (estimated) optimal combination weights.

The regularized regression mentioned in step 3 is discussed in Section 5.

Remark B. A key advantage of the simple linear combination estimator is that the minimizers of the bias-corrected prediction error (10) can be written down explicitly. In Section 8 we discuss more general tuning parameter selection problems in which this will typically not be possible, so that the proposed procedure may not be computationally feasible.

Remark C. Suppose that each $c_{\mathbf{z}}^k(\mathbf{x}_i)$ is a linear least squares fit for a fixed subset of p_k variables. Then it is easy to show that the kth element of the bootstrap correction $\Delta_2(\hat{F})$ is

$$-p_k\hat{\sigma}^2, \quad (14)$$

where $\hat{\sigma}^2$ is the estimated variance of y_i. Thus the bootstrap correction $\Delta_2(\hat{F})$ adjusts $\sum C_{\mathbf{z}}(\mathbf{x}_i)y_i/N$ downward to account for the number of regressors used in each $c_{\mathbf{z}}^k$. The elements of $\Delta_1(\hat{F})$ are more complicated. Let the design matrix for $c_{\mathbf{z}}^k$ be $\mathbf{Z}$, with a corresponding bootstrap value of Z_k^*. Then the kkth element of $\Delta_1(\hat{F})$ is

$$\hat{\sigma}^2\{E_{\hat{F}}[(Z_k^{*T}Z_k^*)^{-1}]Z_k^TZ_k - p\} \geq 0, \quad (15)$$

the inequality following from Jensen's inequality. Thus $\Delta_1(\hat{F})$ will tend to inflate the diagonal of $\sum C_{\mathbf{z}}(\mathbf{x}_i)C_{\mathbf{z}}(\mathbf{x}_i)^T/N$ and hence shrink the least squares estimator $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$. The off-diagonal elements of $\Delta_1(\hat{F})$ are more difficult to analyze; empirically, they seem to be negative when the $c_{\mathbf{z}}^k$'s are positively correlated.

4. LINEAR ESTIMATORS AND GENERALIZED CROSS-VALIDATION

Suppose that each of the estimators $c_{\mathbf{z}}^k(\mathbf{x}_i), i = 1, 2, \ldots, N$, can be written as $\mathbf{H}_k\mathbf{y}$ for some fixed matrix $\mathbf{H}_k$. For example, $\mathbf{H}_k\mathbf{y}$ might be the least squares fit for a fixed subset of the variables $X_1, X_2, \ldots, X_p$, or it might be a cubic smoothing spline fit.

We can obtain an analytic estimate of the combination weights by approximating the cross-validation estimate. Let h_{ii}^k be the iith element of $\mathbf{H}_k$. A standard approximation used in generalized cross-validation (GCV) gives

$$c_{\mathbf{z}(-i)}^k(\mathbf{x}_i) = \frac{c_{\mathbf{z}}^k(\mathbf{x}_i) - y_i \cdot h_{ii}^k}{1 - h_{ii}^k} \approx \frac{c_{\mathbf{z}}^k(\mathbf{x}_i) - y_i \cdot \mathrm{tr}(\mathbf{H}_k)/N}{1 - \mathrm{tr}(\mathbf{H}_k)/N} \equiv \tilde{c}_{\mathbf{z}}^k(\mathbf{x}_i). \quad (16)$$

Therefore, a simple estimate of the combination weights can be obtained by least squares regression of y_i on $\tilde{c}_{\mathbf{z}}^k(\mathbf{x}_i)$. Denote the resulting estimate by $\hat{\boldsymbol{\beta}}_{\mathrm{GCV}}$.

When $c_{\mathbf{z}}^k(\mathbf{x}_i)$ is an adaptively chosen linear fit (e.g., a best subset regression), the foregoing derivation does not apply. However, one might try ignoring the adaptivity in the estimators and use the analytic correction anyway. We explore this idea in the examples of Section 6.3.

5. REGULARIZATION

From the previous discussion, we have four different estimators of the combination weights $\boldsymbol{\beta}$:

- $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$, the least squares estimator defined in (6)
- $\hat{\boldsymbol{\beta}}_{\mathrm{St}}$, the model-mix estimator defined in (8)
- $\hat{\boldsymbol{\beta}}_{\mathrm{Bo}}$, the bootstrap estimator defined in (13)

- $\hat{\beta}_{\mathrm{GCV}}$, the GCV estimator defined below Equation (16), available for linear estimators, $c_{\mathbf{z}}^k(\mathbf{x}) = \mathbf{H}_k\mathbf{y}$.

In our discussion we have derived each of these estimates as minimizers of some roughly unbiased estimate of prediction error $\hat{g}(\mathbf{z}, \hat{F}, \boldsymbol{\beta})$. However, in our simulation study of the next section (and also in Breiman's 1995 study), we find that none of these estimators works well. All of these are unrestricted least squares estimators, and it turns out that some sort of regularization may be needed to improve their performance. This is not surprising, because the same phenomenon occurs in multiple linear regression. In that setting, the average residual sum of squares is unbiased for the true prediction error, but its minimizers—the least squares estimator—does not necessarily possess the minimum prediction error. Often a regularized estimate (e.g., a ridge estimator or a subset regression) has lower prediction error than the least squares estimator.

In terms of our development of Section 2, the prediction error estimate $\hat{g}(\mathbf{z}, F, \boldsymbol{\beta})$ is approximately unbiased for $g(\mathbf{z}, F, \boldsymbol{\beta})$ for *each fixed value* $\boldsymbol{\beta}$, but it is no longer unbiased when a estimator $\hat{\beta}$ is substituted for $\boldsymbol{\beta}$. Roughly unbiased estimators for $g(\mathbf{z}, F, \hat{\beta})$ can be constructed by adding a regularization term to $\hat{g}(\mathbf{z}, F, \boldsymbol{\beta})$ and this leads to regularized versions of the four estimators.

We investigate various forms of shrinkage. Most regularizations can be defined by the addition of a penalty function $J(\boldsymbol{\beta})$ to the corrected prediction error function $\hat{g}(\mathbf{z}, F, \boldsymbol{\beta})$:

$$\tilde{\boldsymbol{\beta}} = \operatorname{argmin}_{\beta}[\hat{g}(\mathbf{z}, F, \boldsymbol{\beta}) + \lambda \cdot J(\beta)], \tag{17}$$

where $\lambda \geq 0$ is a regularization parameter. Let M_{CC} and $M_{C\mathbf{y}}$ be the bias-corrected versions of $\sum_i C_{\mathbf{z}}(\mathbf{x}_i)C_{\mathbf{z}}(\mathbf{x}_i)^T$ and $\sum_i C_{\mathbf{z}}(\mathbf{x}_i)y_i$, obtained by either the bootstrap, cross-validation, or GCV as described earlier. We consider two choices for $J(\boldsymbol{\beta})$. Rather than shrink toward zero (as in ridge regression), it seems somewhat more natural to shrink $\hat{\beta}$ toward $(1/K, 1/K, \ldots, 1/K)^T$, because we might put prior weight $1/K$ on each model fit $c_{\mathbf{z}}^k$. To regularize in this way, we choose $J(\boldsymbol{\beta}) = \|\boldsymbol{\beta} - (1/K)\mathbf{1}\|^2$, leading to the estimate

$$\tilde{\boldsymbol{\beta}} = (M_{CC} + \lambda \cdot I)^{-1}[M_{C\mathbf{y}} + (\lambda/K)\mathbf{1}]. \tag{18}$$

We could choose λ by another layer of cross-validation or bootstrapping, but a fixed choice is more attractive computationally. After some experimentation, we found that a good choice is the value of λ such that the Euclidean distance between $\hat{\beta}$ and $(1/K, \ldots, 1/K)^T$ is reduced by a fixed factor (75%). Some simulations to support this choice are presented in Section 6.

Another regularization forces $\sum \hat{\beta}_i = 1$; this can be achieved by choosing $J(\boldsymbol{\beta}) = \boldsymbol{\beta}^T\mathbf{1} - 1$, leading to

$$\tilde{\boldsymbol{\beta}} = M_{CC}^{-1}[M_{C\mathbf{y}} - \lambda\mathbf{1}], \tag{19}$$

where λ is chosen so that $\tilde{\beta}^T\mathbf{1} = 1$.

To generalize both of these, one might consider estimators of the form $(M_{CC} + \lambda_1 I)^{-1}[M_{C\mathbf{y}} + \lambda_2\mathbf{1}]$ and use a layer of cross-validation or bootstrapping to estimate the best values of λ_1 and λ_2. This is very computationally intensive, and we did not try it in this study.

Still another form of regularization is a nonnegativity constraint, suggested by Breiman (1995). An algorithm for least squares regression under the constraint $\beta_k \geq 0, k = 1, 2, \ldots, K$ was given by Lawson and Hanson (1974), and this can be used to constrain the weights in the model-mix and GCV procedures. To apply this procedure in general, we minimize $\hat{g}(\mathbf{z}, F, \boldsymbol{\beta})$ (Eq. 10) under the constraint $\beta_k \geq 0, k = 1, 2, \ldots, K$, leading to

$$\tilde{\boldsymbol{\beta}} = \operatorname{argmin}_{\beta}(\boldsymbol{\beta}^T M_{CC}\boldsymbol{\beta} - 2M_{C\mathbf{y}}\boldsymbol{\beta}). \tag{20}$$

This problem can be solved by a simple modification of the algorithm given by Lawson and Hansen (1974). In his model-mix procedure, Stone (1974) suggested the constraints $\beta_k \geq 0, k = 1, 2, \ldots, K, \sum \beta_k = 1$. In our simulation experiments, we found that this constraint gave very similar results to the nonnegative constraint, and hence we do not report separate results for it.

Finally, we consider a simple combination method that is somewhat different in spirit than other proposed methods but that constrains $\tilde{\beta}_k \geq 0$ and $\tilde{\beta}^T\mathbf{1} = 1$. We let $\tilde{\beta}_k$ be the relative performance of the kth model. For instance, one might use

$$\tilde{\beta}_k = \frac{L(\mathbf{y}, \hat{\theta}_k, c_{\mathbf{z}}^k)}{\sum_j L(\mathbf{y}, \hat{\theta}_j, c_{\mathbf{z}}^k)}, \tag{21}$$

where $L(\mathbf{y}, \hat{\theta}_k, c_{\mathbf{z}}^k)$ is the maximized likelihood for model k. For a normal model,

$$\tilde{\beta}_k = \frac{\hat{\sigma}_k^{2-n/2}}{\sum_j \hat{\sigma}_j^{2-n/2}},$$

where $\hat{\sigma}_k^2$ is the mean residual error. The estimator can also be motivated as an "estimated posterior mean" where one assumes a uniform prior assigning mass $1/K$ to each model, as we remark later. We replace $\hat{\sigma}_k^2$, the resubstitution estimate of prediction error for model k, with a K-fold cross-validation estimate of prediction error.

Remark D. The relative fit estimator (21) and the other combination estimators of the linear form

$$\sum_{k=1}^{K} \beta_k c_{\mathbf{z}}^k(\mathbf{x}) \tag{22}$$

can be related to the Bayesian formulation for the prediction problem. For instance, the predictive mean of an a new observation Y_0 with a predictor vector $\mathbf{X}_0$ can be expressed as

$$E(Y_0|\mathbf{z}, \mathbf{X}_0) = \int E(Y_0|\mathbf{X}_0, \theta, M_k, \boldsymbol{\beta})p(\theta, \boldsymbol{\beta}, M_k|\mathbf{z})\, d\theta\, d\boldsymbol{\beta}\, dM_k,$$

where M_k represents the kth component model and $\theta = (\theta_1, \ldots, \theta_k)$ represents the parameters corresponding to all component models. The predictive mean can be reexpressed

Table 1. Average Model Errors (and Standard Errors) for Example 1: Many Weak Coefficients

Method	Regularization			
	None	Nonnegativity	Shrinkage	Sum to 1
Least squares	29.3 (1.2)	—	—	—
Ridge (by CV)	25.2 (1.5)	—	—	—
Best subset (by CV)	33.3 (2.2)	—	—	—
Relative fit weights (by CV)	27.6 (1.5)	—	—	—
Combination by least squares	29.3 (1.2)	29.3 (1.2)	29.0 (1.2)	29.3 (1.2)
Combination by CV	69.8 (5.9)	24.0 (1.6)	49.4 (4.2)	66.7 (5.7)
Combination by GCV	60.0 (12.6)	22.4 (1.0)	29.4 (1.4)	30.9 (1.5)
Combination by bootstrap	32.7 (5.6)	21.0 (3.0)	22.2 (3.1)	23.7 (2.9)

as

$$E(Y_0|\mathbf{z},\mathbf{X}_0) = \int\int \left[\sum_k E(Y_0|\mathbf{X}_0,\theta,M_k,\boldsymbol{\beta})p(M_k|\theta,\boldsymbol{\beta},\mathbf{z})\right] \times p(\theta,\boldsymbol{\beta}|\mathbf{z})\,d\theta\,d\boldsymbol{\beta}$$

$$= \int\int \left[\sum_k C(\mathbf{X}_0;\theta,M_k)p(M_k|\theta,\boldsymbol{\beta},\mathbf{z})\right] \times p(\theta,\boldsymbol{\beta}|\mathbf{z})\,d\theta\,d\boldsymbol{\beta},$$

where $C(\mathbf{X}_0;\theta,M_k)$ represents the expected output for model M_k given θ (and $\boldsymbol{\beta}$) at $\mathbf{X}_0$. This formulation also motivates the nonnegative weights (and weights that sum to 1) for the component model outputs. Note that the choice of independent normal priors for β_i corresponds to the ridge-type shrinkage, and a prior consisting of point masses on the coordinate unit vectors corresponds to the relative fit weights.

6. EXAMPLES

6.1 Introduction

In the following examples we compare the combination methods described earlier. Throughout we use ten-fold cross-validation, and $B = 10$ bootstrap samples. Ten-fold cross-validation was found to be superior to leave-one-out cross-validation by Breiman (1995), and it is computationally much less demanding. Estimated model error,

$$\sum_1^N (f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i))^2,$$

is reported; $\hat{f}(\mathbf{x}_i)$ is the estimated regression function and $f(\mathbf{x}_i)$ is the true regression function evaluated at the observed predictor values $\mathbf{x}_i$. We use 25 Monte Carlo simulations in each example, and report the Monte Carlo means and standard deviations of the mean.

6.2 Combining Linear Regression Models

In this example (which is modified from Breiman 1995), we investigate combinations of best subset regression models.

The predictor variables $X_1,\ldots,X_{30}$ were independent standard normal random variables, and the response was generated from the linear model

$$Y = \beta_1 X_1 + \cdots + \beta_{30} X_{30} + \varepsilon,$$

where ε is a standard normal random variable. The β_m were calculated by $\beta_m = \gamma\alpha_m$, where α_m are defined in clusters:

$$\alpha_m = (h - |m-7|)^2 I\{|m-7| < h\} + (h - |m-14|)^2 I\{|m-14| < h\} + (h - |m-21|)^2 I\{|m-21| < h\}.$$

The constant γ was determined so that the signal to noise ratio was approximately equal to 1. Each simulated data set consisted of 60 observations.

Two values, $h = 1$ and $h = 4$, were considered; the case $h = 1$ corresponds to a model with three large nonzero coefficients and $h = 4$ corresponds to a model with many small coefficients. We report the average model errors in Tables 1 and 2.

As expected, for the uncombined methods, ridge regression performs well when there are many small coefficients,

Table 2. Average Model Errors (and Standard Errors) for Example 1: Three Large Coefficients

Method	Regularization			
	None	Nonnegativity	Shrinkage	Sum to 1
Least squares	29.3 (1.2)	—	—	—
Ridge (by CV)	24.1 (.9)	—	—	—
Best subset (by CV)	8.5 (1.5)	—	—	—
Relative fit weights (by CV)	8.7 (1.5)	—	—	—
Combination by least squares	29.3 (1.2)	29.3 (1.2)	28.9 (1.2)	29.3 (1.2)
Combination by CV	45.3 (3.6)	9.5 (1.1)	28.1 (2.7)	36.7 (2.7)
Combination by GCV	52.5 (18.2)	14.0 (1.0)	27.0 (1.1)	29.5 (1.4)
Combination by bootstrap	16.5 (2.19)	10.1 (1.0)	11.3 (1.1)	12.6 (1.2)

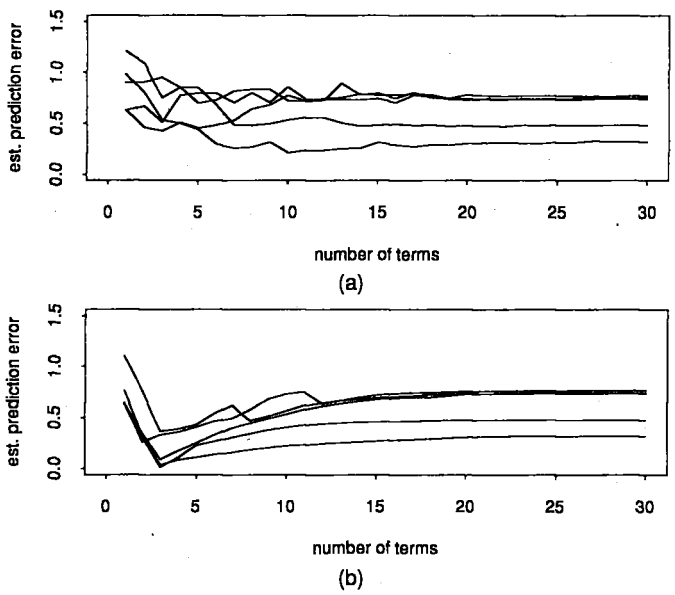

Figure 1. Estimated Model Errors for Best Subsets for Five Realizations of Each of the Linear Models. (a) Many weak coefficients; (b) three large coefficients.

and best subset regression is the clear choice when there were a few large coefficients.

Most of the combination methods without regularization do not yield smaller model errors than standard methods; the cross-validation combination method and GCV method give substantially larger model errors than ordinary least squares, ridge regression, or best subsets. Only the bootstrap method for the three large coefficient model yields smaller model errors than ordinary least squares and ridge regression.

However, regularization substantially reduces the model error of the combination methods. The nonnegative estimators seem to perform as well as, and sometimes far better than, the shrinkage estimators and the estimators constrained to sum to 1. The bootstrap and the cross-validation combination methods with nonnegativity constraints yield smaller average model errors than ridge regression and ordinary least squares for both the many-small-coefficient model and the three-large-coefficient model and yield results very close to best subsets for the three-large-coefficient model.

The relative fit weights seem to perform well relative to the other combination methods for the three-large-coefficient model, but yield somewhat larger errors than some of the combination methods with regularization for the model with many small coefficients.

Overall, combination methods are more helpful in reducing model error over best subset regression and sometimes ridge regression for the many-small-coefficients model. The improved performance of the combination methods in this case is likely due to instability of best subset selection for the many-small-coefficients model. Figure 1 shows the estimated model errors for five simulated data sets for the sequence of best subset regressions. The complexity of the best subset model would be much less stable in the many-small-coefficient model compared to the three-large-coefficient model. It is clear that any combination method would likely not help for the three-large-coefficient model.

6.3 Boston Housing Data

We applied the combination methods to the Boston housing data of Harrison and Rubinfield (1978). The response variable is the median price of owner-occupied homes in 1,000's of dollars in census tracts in Boston and the surrounding area. The 13 predictor variables include demographic, economic, and land use variables. There are 506 observations in the data set. We used a random sample of one-third of the observations to develop a regression model and the remaining approximately two-thirds of the observations to estimate the prediction error of the procedures. We generated 50 random learning and test samples, and we present mean prediction errors over the 50 samples.

For this example, we expect a smaller improvement for combination methods with regularization over either ridge regression or subset selection, because in this case there are a large number of observations and not a large number of predictors, and several of the predictors are strongly associated with the response. In favor of combination methods, some of the predictors are at least quite highly correlated, which introduces some instability into the models selected by a best subsets regression method.

The estimated prediction errors are presented in Table 3. The prediction error for the bootstrap combination method with nonnegative coefficients is smaller than the best model selected by ten-fold cross-validation. The least squares model also performs as well the combination or ridge method, because there is little problem with overfitting, due to the relatively large data set and several strong predictors of house prices. The standard errors of predic-

Table 3. Average Prediction Errors (and Standard Errors) for Boston Housing Data

	Regularization			
Method	*None*	*Nonnegativity*	*Shrinkage*	*Sum to 1*
Least squares	26.3 (.44)	—	—	—
Ridge (by CV)	26.3 (.44)	—	—	—
Best subset (by CV)	26.8 (.53)	—	—	—
Relative fit weights (by CV)	26.3 (.44)	—	—	—
Combination by least squares	26.3 (.44)	26.3 (.44)	26.0 (.44)	26.3 (.44)
Combination by CV	30.7 (.84)	26.7 (.49)	28.2 (.56)	30.6 (.95)
Combination by GCV	30.3 (.76)	26.5 (.48)	27.4 (.54)	30.1 (.68)
Combination by bootstrap	29.7 (2.5)	26.3 (.44)	26.4 (.48)	29.7 (2.5)

Table 4. Average Prediction Errors (and Standard Errors) for Boston Housing Data With Extra Noise Predictors and a Smaller Training Sample

Method	Regularization			
	None	Nonnegativity	Shrinkage	Sum to 1
Least squares	36.4 (.92)	—	—	—
Ridge (by CV)	38.5 (1.1)	—	—	—
Best subset (by CV)	35.1 (.83)	—	—	—
Relative fit weights (by CV)	32.2 (.60)	—	—	—
Combination by least squares	36.4 (.92)	36.4 (.92)	35.9 (.89)	36.4 (.92)
Combination by CV	53.2 (3.1)	32.2 (.61)	42.4 (1.8)	52.0 (3.1)
Combination by GCV	44.7 (3.1)	31.8 (.67)	37.3 (1.6)	41.8 (2.6)
Combination by bootstrap	34.4 (1.2)	31.2 (.47)	31.6 (.60)	34.2 (1.2)

tions appear quite large relative to the difference observed, but the standard errors of paired differences in prediction errors indicate that combination by bootstrap with nonnegative coefficients or by shrinkage perform better than combination methods without regularization or combination methods, which just force the coefficients to sum to 1. However, only a small improvement is seen over best subset selection.

Therefore, in this example with a relatively small number of important predictors and a relatively large sample size, the combination method by bootstrap with nonnegative coefficients performed only slightly better than best subset selection and performed about well as ridge regression.

However, from the simulation studies, we would expect a larger improvement by bootstrap combination with nonnegative coefficients if there were a larger number of weakly predictive covariates and/or a smaller training set size. Therefore, to test this hypothesis, we increased the difficulty of the Boston housing data prediction problem by both reducing the training sample to one-half of the training set size used earlier. In addition, we doubled the predictors by adding 13 more noise predictors, standard normal distributed but unrelated to housing prices. As before, we repeated the analysis for 50 randomly generated training and test samples; we report the mean prediction errors in Table 4.

In this case the bootstrap combination with nonnegative coefficients performs substantially better than ridge regression or best subset selection with 18% and 11% improvements in estimated prediction error. Almost as great improvements were obtained by bootstrap combination with shrinkage and combination by cross-validation and GCV with nonnegative coefficients.

6.4 Extensions to Other Regression Models

There is little difficulty in extending the techniques to combinations of more complicated regression predictors. GCV-type combination may not be feasible for methods that adaptively select basis functions or transformations of the predictor variables, but K-fold cross-validation and bootstrap methods can be directly applied. For instance, we have experimented with combining a sequence of tree-based derived by the cost-complexity pruning algorithm of the classification and regression tree (CART) algorithm (Breiman, Friedman, Olshen, and Stone 1984). Often the selection the size of the best tree in sequence trees by cross-validation is quite unstable, and the number of pruned trees in the sequence can be quite large. Therefore, there is potential for combination methods to reduce prediction error in this setting. We have observed this in our limited simulation experiments.

Table 5. Average Model Errors (and Standard Errors)

No. of bootstrap samples	Model errors
5	20.63 (1.43)
10	20.14 (1.39)
25	20.15 (1.46)
50	19.98 (1.36)

Remark E. There is an open question of how many bootstrap samples is sufficient for the procedure to work well. We repeated the simulation experiment for the regression model with many small coefficients given in Section 6 with 5, 10, 25, and 50 bootstrap samples. The mean model errors for the bootstrap combination method with nonnegativity constraints are given in Table 5.

There appears to be little improvement in the reduction of model error beyond 10 bootstrap samples.

Remark F. The shrinkage parameter was set so that the Euclidean distance between the estimated coefficients and the vector $(1/K, \ldots, 1/K)^T$ was reduced by 75%. We chose a fixed factor for the shrinkage parameter for the simulations because it was most attractive computationally and it was similar to the other regularization methods, because no additional calibration was required. To investigate the choice of 75%, we repeated the simulation experiments for three fixed values of the shrinkage parameters corresponding to 75%, 50%, and 25% reduction in the Euclidean distance between the estimated coefficients and the vector $(1/K, \ldots, 1/K)^T$. We compared the fixed choices to the best choice of the shrinkage parameter among those three values for that estimated model. The mean of the ratios of the model errors for the fixed reduction to the model error to best choice of shrinkage parameter are presented in Tables 6 and 7. For both types of variable effects, the 75% shrinkage performs best among these three fixed choices. In addition, the model error for the fixed 75% choice is at most 2% greater than the adaptive choice and in most cases is much closer to the best adaptive value. However, one would expect some further reductions in model error

Table 6. Mean of the Ratios of Model Errors for Fixed Amounts of Shrinkage to Model Errors for the Best Adaptive Shrinkage (Standard Errors) for Example 1: Many Weak Coefficients

	Regularization		
	75%	50%	25%
Combination by least squares	1.0011 (.0008)	1.0125 (.0022)	1.0125 (.0023)
Combination by CV	1.0000 (.0000)	1.1955 (.0295)	1.3152 (.0469)
Combination by GCV	1.0048 (.0033)	1.0730 (.0297)	1.1728 (.0769)
Combination by bootstrap	1.0065 (.0035)	1.1363 (.0993)	1.3092 (.2349)

by selecting the best shrinkage parameter among a larger number of potential values.

7. CLASSIFICATION PROBLEMS

In a classification problem, the outcome is not continuous but rather falls into one of J outcome classes. We can view this as a regression problem with a multivariate J-valued response y having a 1 in the jth position if the observation falls in class j and 0 otherwise.

Most classification procedures provide estimated class probabilities functions $\hat{\mathbf{p}}(\mathbf{x}) = (\hat{p}_1(\mathbf{x}), \ldots, \hat{p}_J(\mathbf{x}))^T$. Given K such functions, $\mathbf{p}^k(\mathbf{x}) = (\hat{p}_1^k(\mathbf{x}), \ldots, \hat{p}_J^k(\mathbf{x}))^T, k = 1, 2, \ldots, K$, it seems reasonable to apply the combination strategies to either the probabilities themselves or their log ratios. That is, we take $c_{\mathbf{z}}^{kj}(\mathbf{x})$, the jth component of $c_{\mathbf{z}}^k(\mathbf{x})$, to be either

$$c_{\mathbf{z}}^{kj}(\mathbf{x}) = \hat{p}_j^k(\mathbf{x}) \tag{23}$$

or

$$c_{\mathbf{z}}^{kj}(\mathbf{x}) = \log \frac{\hat{p}_j^k(\mathbf{x})}{\hat{p}_J^k(\mathbf{x})}. \tag{24}$$

Table 7. Mean of the Ratios of Model Errors for Fixed Amounts of Shrinkage to Model Errors for the Best Adaptive Shrinkage (Standard Errors) for Example 1: Several Large Coefficients

	Regularization		
	75%	50%	25%
Combination by least squares	1.0000 (.0008)	1.0121 (.0014)	1.0126 (.0014)
Combination by CV	1.0002 (.0002)	1.3014 (.0404)	1.5019 (.0753)
Combination by GCV	1.0035 (.0033)	1.0748 (.0130)	1.1216 (.0243)
Combination by bootstrap	1.0188 (.0142)	1.0715 (.0153)	1.1113 (.0240)

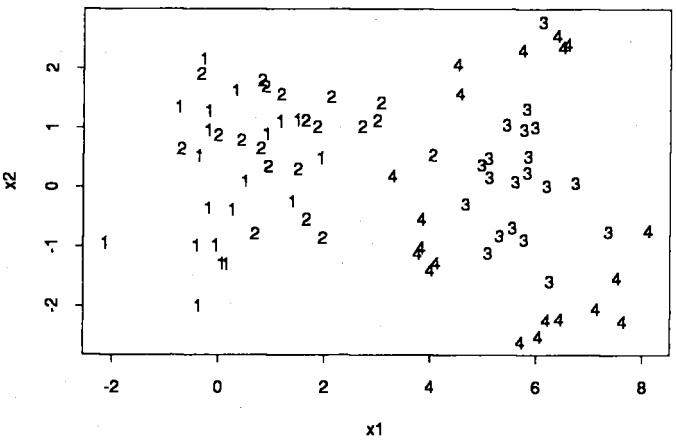

Figure 2. Typical Realization for Example 2.

In our experiments, we found that the use of probabilities (23) gives better classification results and is more interpretable.

Let $\hat{\mathbf{P}}(\mathbf{x}) = (\hat{p}_1^1(\mathbf{x}), \hat{p}_2^1(\mathbf{x}), \ldots, \hat{p}_J^1(\mathbf{x}), \hat{p}_1^2(\mathbf{x}), \hat{p}_2^2(\mathbf{x}), \ldots)^T$, a vector of length JK. Then the combination estimator has the form

$$\theta_j(\mathbf{x}) = \boldsymbol{\beta}_j^T \hat{\mathbf{P}}(\mathbf{x}); \qquad j = 1, 2, \ldots, J, \tag{25}$$

where $\boldsymbol{\beta}_j$ is a JK vector of weights. We then apply the straightforward multivariate analogs of the procedures described earlier. The population regression problem (4) becomes a multiresponse regression, producing vector-valued estimates $\hat{\boldsymbol{\beta}}_j^T$. The combination estimators are computed in an analogous fashion to the regression case. Finally, given the combination estimators $\hat{\theta}_j(\mathbf{x}) = \hat{\boldsymbol{\beta}}_j^T \hat{\mathbf{P}}(\mathbf{x})$, the estimated class probabilities are taken to be $\hat{\theta}_j(\mathbf{x}) / \sum_j \hat{\theta}_j(\mathbf{x})$. A nonnegativity constraint for the weights is especially attractive here, because it ensures that these estimated probabilities lie between 0 and 1.

7.1 Combining Linear Discriminant Analysis and Nearest Neighbors

In this example the data are two-dimensional, with four classes. A typical data realization is shown in Figure 2. Classes 1, 2, and 3 are standard independent normal, centered at (0, 0), (1, 1), and (6, 0). Class 4 was generated as standard independent normal with mean (6, 0), conditional on $4.5 \leq x_1^2 + x_2^2 \leq 8$. There were 10 observations in each class.

We consider combination of linear discriminant analysis (LDA) and the 1-nearest-neighbor method. The idea is that

Table 8. Average % Test Error (and Standard Deviations) for Example 3

	Regularization	
Method	None	Nonnegativity
LDA	36.3 (.64)	—
1-nearest-neighbor	27.5 (.87)	—
Combination by least squares	26.1 (.94)	25.2 (.90)
Combination by bootstrap	27.0 (.95)	25.7 (.95)

Table 9. Average Combination Weights for Example 3

	LDA				1-nearest-neighbor			
Class	1	2	3	4	1	2	3	4
1	.75	0	0	0	.31	0	0	0
2	0	.80	0	0	0	.29	0	0
3	0	0	.02	.07	0	0	.95	0
4	0	0	.01	.01	0	0	0	1.13

LDA should work best for classes 1 and 2 but not for classes 3 and 4. Nearest-neighbor should work moderately well for classes 1 and 2 and much better than LDA for classes 3 and 4. By combining the two methods, we might be able to improve on both.

To proceed, we require from each method a set of estimated class probabilities for each observation. For LDA, we used the estimated Gaussian probabilities; for 1-nearest-neighbor, we estimated the class j probability for feature vector $\mathbf{x}$ by $\exp(-d^2(\mathbf{x},j))/\sum_{j=1}^{J}\exp(-d^2(\mathbf{x},j))$, where $d^2(\mathbf{x},j)$ is the squared distance from $\mathbf{x}$ to the nearest neighbor in class j. The results of 10 simulations of the combined LDA/nearest-neighbor procedure are shown in Table 8.

We see that simple least squares combination, or regularized combination by bootstrap, slightly improves on the LDA and 1-nearest-neighbor rules. More interesting are the estimated combination weights produced by the nonnegativity constraint. Table 9 shows the average weights from the combination by bootstrap (the combination by least squares weights are very similar). LDA gets higher weight for classes 1 and 2, whereas 1-nearest-neighbor is used almost exclusively for classes 3 and 4, where the structure is highly nonlinear.

In general, this procedure might prove to be useful in determining which outcome classes are linearly separable and which are not.

8. MORE GENERAL COMBINATION SCHEMES

More general combination schemes would allow the β_k's to vary with $\mathbf{x}$:

$$\sum_{1}^{K}\beta_k(\mathbf{x})c_{\mathbf{z}}^k(\mathbf{x}). \tag{26}$$

A special case of this would allow β_k to vary only with $c_{\mathbf{z}}^k$; that is,

$$\sum_{1}^{K}\beta_k(c_{\mathbf{z}}^k(\mathbf{x}))c_{\mathbf{z}}^k(\mathbf{x}). \tag{27}$$

Both of these models fall into the category of the "varying coefficient" model discussed by Hastie and Tibshirani (1993). The difficulty in general is how to estimate the functions $\beta_k(\cdot)$; this might be easier in model (27) than in model (26), because the $c_{\mathbf{z}}^k$'s are all real valued and are probably of lower dimension than $\mathbf{x}$.

One application of varying combination weights would be in scatterplot smoothing. Here each $c_{\mathbf{z}}^k(x)$ is a scatterplot smooth with a different smoothing parameter λ_k. The simple combination $\sum\beta_k c_{\mathbf{z}}^k(x)$ gives another family of estimators that are typically outside of the original family of estimators indexed by λ. However, it would seem more promising to allow each β_k to be a function of x and hence allow local combination of the estimates. Requiring $\beta_k(x)$ to be a smooth function of x would ensure that the combination estimator is also smooth.

Potentially, one could use the ideas here for more general parameter selection problems. Suppose that we have a regression estimator denoted by $\eta_{\mathbf{z}}(\mathbf{x},\boldsymbol{\beta})$. The estimator is computed on the data set $\mathbf{z}$, with an adjustable (tuning) parameter vector $\boldsymbol{\beta}$ and gives a prediction at $\mathbf{x}$. We wish to estimate the value of $\boldsymbol{\beta}$ giving the smallest prediction error,

$$g(\mathbf{z},F,\boldsymbol{\beta}) = E_{0F}(Y_0 - \eta_{\mathbf{z}}(\mathbf{X}_0,\boldsymbol{\beta}))^2. \tag{28}$$

Then we can apply the bootstrap technique of Section 3 to estimate the bias in $g(\mathbf{z},\hat{F},\boldsymbol{\beta}) = \sum_{i=1}^{N}(y_i - \eta_{\mathbf{z}}(\mathbf{x}_i,\boldsymbol{\beta}))^2$. Using the notation of Section 3, the biased corrected estimator has the form

$$\hat{g}(\mathbf{z},\hat{F},\boldsymbol{\beta}) = \sum_{i=1}^{N}(y_i - \eta_{\mathbf{z}}(x_i,\hat{\boldsymbol{\beta}}))^2 + \hat{\Delta}(\hat{F},\boldsymbol{\beta}), \tag{29}$$

where

$$\hat{\Delta}(\hat{F},\boldsymbol{\beta}) = \frac{1}{B}\left[\frac{1}{N}\sum_{i=1}^{N}(y_i - \eta_{\mathbf{z}^{*b}}(\mathbf{x}_i,\boldsymbol{\beta}))^2 - \frac{1}{N}\sum_{i=1}^{N}(y_i^{*h} - \eta_{\mathbf{z}^{*b}}(\mathbf{x}_i^{*b},\boldsymbol{\beta}))^2\right].$$

We would then minimize $\hat{g}(\mathbf{z},\hat{F},\boldsymbol{\beta})$ over $\boldsymbol{\beta}$. This idea may be useful only if $\eta_{\mathbf{z}}(\mathbf{x},\boldsymbol{\beta})$ can be written as an explicit function of $\boldsymbol{\beta}$, so that $\hat{g}(\mathbf{z},\hat{F},\boldsymbol{\beta})$ and its derivatives can be easily computed. In this article we have considered the linear estimator $\eta_{\mathbf{z}}(\mathbf{x},\boldsymbol{\beta}) = \sum\beta_k c_{\mathbf{z}}^k(\mathbf{x})$ for which the minimizer of $\hat{g}(\mathbf{z},\hat{F},\boldsymbol{\beta})$ can be explicitly derived by least squares. The variable kernel estimator of Lowe (1993) is another example where this procedure could be applied: in fact, Lowe uses the cross-validation version of the foregoing procedure to estimate the adjustable parameters $\boldsymbol{\beta}$. In many adaptive procedures (e.g., tree-based regression), the estimator $\eta_{\mathbf{z}}(\mathbf{x},\boldsymbol{\beta})$ is a complicated function of its tuning parameters, so that minimization of $\hat{g}(\mathbf{z},\hat{F},\boldsymbol{\beta})$ would be quite difficult.

9. DISCUSSION

This investigation suggests that combining of estimators, used with some regularization, can be a useful tool both for improving prediction performance and for learning about the structure of the problem. We have derived and studied a number of procedures and found that cross-validation (model-mix) and the bootstrap, used with a nonnegativity constraint, seemed to work best. The bootstrap estimates seem to require far less regularization; the reason for this is not clear.

An interesting question for further study is: How can we choose estimators for which combining will be effective for a given problem? We expect that combining will be most useful when the underlying model is best represented by a combination of two or more models where good methods exist for prediction for each of the models, such as the classification example described in Section 7. In addition, recent work by Breiman (1996) also suggests that combining models of different complexity but in the same class, such as best $2, 3, 4, \ldots, k$ predictor regressions models, will be helped most when selection of the best model is unstable; this was seen in the linear regression simulation examples in Section 6. The application of combining where selection of a best model selection is unstable also supports using combination-type methods for tree-based methods or neural networks where the choice of best-fitting model is often unstable.

[Received March 1994. Revised May 1996.]

REFERENCES

Breiman, L. (1995), "Stacked Regressions," *Machine Learning*, 24, 49–64.

Breiman, L. (1996), "Bagging Predictors", *Machine Learning*, 26, 123–140

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984), *Classification and Regression Trees,* Pacific Grove, CA: Wadsworth.

Clark, L., and Pregibon, D. (1992), "Tree-Based Models," in *Statistical Models in S,* eds. J. M. Chambers and T. Hastie, Pacific Grove, CA: Wadsworth.

Efron, B. (1983), "Estimating the Error Rate of a Prediction Rule: Some Improvements on Cross-Validation," *Journal of the American Statistical Association,* 78, 316–331.

Efron, B., and Tibshirani, R. (1993), *An Introduction to the Bootstrap,* New York: Chapman and Hall.

Harrison, D., and Rubinfeld, D. L. (1978), "Hedonic Prices and the Demand for Clean Air," *Journal of Environmental Economics and Management,* 5, 81–102.

Hastie, T. J., and Pregibon, D. (1991), "Shrinking Trees," technical report, AT&T Bell Laboratories.

Hastie, T. J., and Tibshirani, R. J. (1993), "Varying Coefficient Models" (with discussion), *Journal of the Royal Statistical Society,* Ser. B, 55, 757–796.

Lawson, C. L., and Hanson, R. J. (1974), *Solving Least Squares Problems,* Englewood Cliffs, NJ: Prentice-Hall.

Lowe, D. G. (1993), "Similarity Metric Learning for a Variable Kernel Classifier," technical report, University of British Columbia, Dept. of Computer Science.

McCullagh, P., and Tibshirani, R. (1988), "A Simple Adjustment for Profile Likelihoods," *Journal of the Royal Statistical Society,* Ser. B, 52(2), 325–344.

Stone, M. (1974), "Cross-Validation Choice and Assessment of Statistical Predictions" (with discussion), *Journal of the Royal Statistical Society,* Ser. B, 36, 111–147.

Wolpert, D. (1992), "Stacked Generalization," *Neural Networks,* 5, 241–259.