Machine Learning, 24, 49–64 (1996)
© 1996 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Stacked Regressions

LEO BREIMAN leo@stat.berkeley.edu
Statistics Department, University of California, Berkeley, CA 94720

Editor: Ross Quinlan

Abstract. Stacking regressions is a method for forming linear combinations of different predictors to give improved prediction accuracy. The idea is to use cross-validation data and least squares under non-negativity constraints to determine the coefficients in the combination. Its effectiveness is demonstrated in stacking regression trees of different sizes and in a simulation stacking linear subset and ridge regressions. Reasons why this method works are explored. The idea of stacking originated with Wolpert (1992).

Keywords: Stacking, Non-negativity, Trees, Subset regression, Combinations

1. Introduction

Suppose one has available K predictors $v_1(\boldsymbol{x}), \ldots, v_K(\boldsymbol{x})$ of a numerical outcome variable y in terms of a vector $\boldsymbol{x}$. Assume these were constructed using the same learning set $\mathcal{L} = \{(y_n, \boldsymbol{x}_n),\ n = 1, \ldots, N\}$ where each $\boldsymbol{x}_n$ is an input vector. In common examples the predictors are of the same type but differ in complexity. For instance, in CART one takes $v_k(\boldsymbol{x})$ to be the predictor using the subtree with k terminal nodes. Another example is in linear regression. If $\boldsymbol{x}$ is K-dimensional, then $v_k(\boldsymbol{x})$ is that least squares regression based on k variables having lowest learning set squared error. Other examples occur when the predictors are derived using different methods, i.e. neural nets, nearest-neighbor, etc.

What is wanted is not a collection of predictors, but a single predictor. Often, the approach used is to find the single best predictor among the $\{v_k(\boldsymbol{x})\}$. This is done either using a test set or cross-validation to estimate the prediction error of each $v_k(\boldsymbol{x})$ on "future" data.

David Wolpert (1992), in an article appearing in the Neural Network literature, proposed the following interesting idea: if we have a set of predictors (linear or nonlinear) $v_1(\boldsymbol{x}), \ldots, v_K(\boldsymbol{x})$ then instead of selecting a single one from this set, a more accurate predictor can be gotten by combining the $v_1, \ldots, v_K$. The method for combination is based on *level one* data defined as follows: leave out the nth case and repeat the procedures for constructing the predictors, getting $v_k^{(-n)}(\boldsymbol{x})$, $k = 1, \ldots, K$. Define the K-variable vector $\boldsymbol{z}_n$ by

$$z_{kn} = v_k^{(-n)}(\boldsymbol{x}_n).$$

Then the level one data consists of $\{(y_n, \boldsymbol{z}_n),\ n = 1, \ldots, N\}$.

Ordinarily the level one data would be used to select one of the v_k, i.e. use v_k where k minimizes $\sum_k (y_n - z_{kn})^2$. Wolpert's idea is that the level one data has more information in it, and can be used to construct "good" combinations of the v_{kn}. He illustrated this idea by using level one data to form nonlinear combinations of nearest neighbor predictors. But he also remarks that just how to use level one data to form accurate combinations is "black art".

The idea of combining predictors instead of selecting the single best is well-known in statistics and has a long and honorable theoretical background. We include several references—Rao and Subrahmaniam (1971), Efron and Morris (1973), Rubin and Weisberg (1975), Berger and Bock (1976), and Green and Strawderman (1991). What is not spelled out is how to combine given a finite data set.

We simplify the problem by restricting attention to combinations of the form

$$v(\boldsymbol{x}) = \sum_k \alpha_k v_k(\boldsymbol{x}).$$

Given a learning set $\mathcal{L} = \{(y_n, \boldsymbol{x}_n), n = 1, \ldots, N\}$ one possibility is to take the $\{\alpha_k\}$ to minimize

$$\sum_n (y_n - \sum_k \alpha_k v_k(\boldsymbol{x}_n))^2 \tag{1.1}$$

There are two problems with this approach. The first is that if the $v_k(\boldsymbol{x})$ were constructed using $\mathcal{L}$, and the $\{\alpha_k\}$ gotten by minimizing squared error over $\mathcal{L}$, then the resulting $\{\alpha_k\}$ will overfit the data – generalization will be poor. This problem can be fixed by using the level-one cross-validation data. That is, take the $\{\alpha_k\}$ to minimize

$$\sum_n (y_n - \sum_k \alpha_k z_{kn})^2 \tag{1.2}$$

where $z_{kn} = v_k^{(-n)}(\boldsymbol{x}_n)$.

The second problem is more difficult. Usually the $\{v_k(\boldsymbol{x})\}$ are strongly correlated, since they are all trying to predict the same thing. If the $\{\alpha_k\}$ are computed as the minimizers of (1.1) or its cross-validation version (1.2), then they will be highly sensitive to small changes in the data. Again, generalization will be poor.

The usual method for estimating regression coefficients of highly correlated variables is ridge regression (Hoerl and Kennard (1970)). This is the recipe that minimizes

$$\sum_n (y_n - \sum_k \alpha_k z_{kn})^2$$

under the constraint $\sum \alpha_k^2 = s$. Then the optimal value of s is selected by another cross-validation. In our experiments, ridge and some variants of ridge were tried. Results were better than using the least squares $\{\alpha_k\}$, but were not consistent.

A method that gives consistently good results is this: minimize

$$\sum_n (y_n - \sum_k \alpha_k z_{kn})^2$$

under the constraints $\alpha_k \geq 0$, $k = 1, \ldots, K$. This resulting predictor $\sum \alpha_k v_k$ appears to almost always have lower prediction error than the single prediction v_k having lowest cross-validation error.

The word "appears" is used because a general proof is not yet in place. The evidence offered consists of some analytic work in Section 2, experiments on trees and extensive simulations. The lowering of error rates is pervasive. Because Wolpert named his method "stacked generalizations" we refer to the present method as *stacked regressions*. The plan for the presentation of the results is as follows: In Section 2 we give some general reasons why this method works as well as it does. In Section 3, the method is applied to stacking trees of different sizes. Using this method on two well-known data bases results in a 10% reduction in error on each.

Statisticians often (and myself in particular) use linear regression prediction as mouse experiments on new methods. Some reasons are that it is easy to generate simulated data, to compute the regressions and to carry out large simulations. Sections 4-8 cover simulation experiments with linear regression. In Section 4 we give some basic definitions and define the simulation structure: 40 input variables and 60 cases are used. In Section 5, the results of stacking subset regressions are examined. That is, the $v_1, \ldots, v_K$ are the subset regressions gotten by stepwise backward deletion of variables. Section 6 looks at stacking ridge regressions, i.e. the $v_1, \ldots, v_K$ are the ridge regressions corresponding to ridge parameters $\lambda_1, \ldots, \lambda_K$.

In Section 7, the subset regressions and ridge regressions are stacked together and the results compared to selecting (via cross-validation) the best of the best subset regression and best ridge regression. Stacking wins, sometimes by a large margin.

Generating the level one data by leave-one-out cross-validation can be very computer intensive. In Section 8 simulation evidence is used to show that (surprizingly) the level one data generated by much cheaper 10-fold cross-validation is more effective than the leave-one-out level one data. Section 9 contains various concluding comments concerning the stacked procedures, and Section 10 gives conclusions.

The results show that stacking can produce predictors with substantially reduced prediction errors. Stacking never does worse than selecting the single best predictor. We note that the biggest gains came when dissimilar sets of predictors were stacked, i.e. subset selection and ridge regressions. The more similar the predictors, the less advantage there is in stacking. We have also experimented in stacking linear regression predictors with k-nearest neighbor predictors. Here again, there were substantial reductions in error.

Stimulated by my technical report on stacking, Le Blanc and Tibshirani (1993) investigated other methods of stacking, but also come to the conclusion that non-negativity constraints lead to the most accurate combinations. They also gave examples where stacking classifiers gives increased accuracy. Perrone (1994) has also done relevant work.

2. Why Non-negativity Constraints Work

Only partial answers are available. Suppose that the $\{v_k(x)\}$ are strongly correlated and the $\{\alpha_k\}$ are chosen using least squares or ridge regression. Then there is no guarantee that the resulting predictor $\sum_k \alpha_k v_k(x)$ will stay near the range $[\min_k v_k(x), \max_k v_k(x)]$ and generalization may be poor.

Now consider imposing the non-negativity constraints on the $\{\alpha_k\}$ together with the additional constraint $\sum \alpha_k = 1$. For any $\{\alpha_k\}$ satisfying the constraints $\alpha_k \geq 0$, $\sum_k \alpha_k = 1$,

$$v(\boldsymbol{x}) = \sum_k \alpha_k v_k(\boldsymbol{x})$$

is an "interpolating" predictor. That is, for every value of $\boldsymbol{x}$,

$$\min_k v_k(\boldsymbol{x}) \leq v(\boldsymbol{x}) \leq \max_k v_k(\boldsymbol{x}).$$

So, what our procedure does is to find the best "interpolating" predictor.

To do some more exploration, note that

$$\sum_n (y_n - \sum_k \alpha_k v_k(\boldsymbol{x}_n))^2 = \sum_{i,j} \alpha_i \alpha_j R_{ij}$$

where R_{ij} is the matrix of residual products

$$R_{ij} = \sum_n (y_n - v_i(\boldsymbol{x}_n))(y_n - v_j(\boldsymbol{x}_n)).$$

Now suppose we want to determine the $\{\alpha_k\}$ as the minimizers of $\boldsymbol{\alpha}^t R \boldsymbol{\alpha}$ under the non-negativity and sum one constraints (the cross-validation data is used in practice to determine the $\{\alpha_k\}$ but we ignore this to present the conclusions in a simple form). An important question is: under what conditions on the matrix R is the best single predictor also the best stacked predictor?

By the best single predictor we mean that v_k such that $R_{kk} = \min_j R_{jj}$. Then the following holds:

THEOREM 1 *The best single predictor v_k is also the best stacked predictor if and only if*

$$R_{kk} \leq R_{ik}, \quad all \ i.$$

Proof: Since the minimization of $\boldsymbol{\alpha}^t R \boldsymbol{\alpha}$ under $\boldsymbol{\alpha} \geq 0$, $\sum \alpha_k = 1$ is a quadratic programming problem, the Kuhn-Tucker conditions are necessary and sufficient for a solution (see, for example, Luenberger (1984)). In this case the conditions are that there is a λ, and vector $\boldsymbol{\mu}$ such that

$$R\boldsymbol{\alpha} = \lambda + \boldsymbol{\mu}$$

where $\alpha_k > 0 \Rightarrow \mu_k = 0$ and $\alpha_k = 0 \Rightarrow \mu_k \geq 0$. If $\alpha_j = \delta_k(i)$ is the solution, then

$$R_{ik} = \lambda + \mu_i$$

implying that $R_{kk} = \lambda$ and $R_{ik} \geq R_{kk}$. Conversely, suppose that $R_{kk} \leq R_{ik}$. Then for $\alpha_i = \delta_k(i)$, $\sum_i R_{ji}\delta_k(i) = R_{jk}$. Putting $\lambda = R_{kk}$ gives $R\boldsymbol{\alpha} = \lambda + \boldsymbol{\mu}$ where $\boldsymbol{\mu}$ has the requisite properties. ■

Let ρ_{ik} be the correlation between the ith and kth set of residuals and σ_i, σ_k their standard deviations. The best single prediction is the best stacking predictor iff

$$\frac{\sigma_k}{\sigma_i} \le \rho_{ik}.$$

Thus, if $\sigma_i \simeq \sigma_k$ then $\rho_{ik} \simeq 1$. Put another way, all $v_i(\boldsymbol{x})$ with comparable squared error to $v_k(\boldsymbol{x})$ have to be nearly equal to $v_k(\boldsymbol{x})$.

Although the above remarks are true only if $\sum \alpha_k = 1$, it turns out that this constraint is largely unnecessary. That is, the sum-unconstrained optimum $\{\alpha_k\}$ will have the property that $\sum_k \alpha_k \simeq 1$. To see this, let the non-negative sum-unconstrained minimizer of

$$\sum_n (y_n - \sum \alpha_k v_k(\boldsymbol{x}_n))^2 \tag{2.1}$$

have sum s, and put $\tilde{\alpha}_k = \alpha_k/s$, $v = \sum \alpha_k v_k$ and $v^* = \sum \tilde{\alpha}_k v_k$.

For simplicity, we use the notation $\|\boldsymbol{\mu}\|^2 = \sum_n \mu_n^2$. Then

$$\begin{aligned} \|\boldsymbol{y} - \boldsymbol{v}\| &= \|(1-s)\boldsymbol{y} + s(\boldsymbol{y} - \boldsymbol{v}^*)\| \\ &\ge |1-s|\,\|\boldsymbol{y}\| - s\|\boldsymbol{y} - \boldsymbol{v}^*\| \end{aligned}$$

leading to

$$\frac{\|\boldsymbol{y}\| - \|\boldsymbol{y} - \boldsymbol{v}\|}{\|\boldsymbol{y}\| + \|\boldsymbol{y} - \boldsymbol{v}^*\|} \le s \le \frac{\|\boldsymbol{y}\| + \|\boldsymbol{y} - \boldsymbol{v}\|}{\|\boldsymbol{y}\| - \|\boldsymbol{y} - \boldsymbol{v}^*\|}.$$

If there are some good predictors among the $\{v_k\}$ then both $\|\boldsymbol{y} - \boldsymbol{v}\|$ and $\|\boldsymbol{y} - \boldsymbol{v}^*\|$ will be small compared to $\|\boldsymbol{y}\|$ and the minimizing $\{\alpha_k\}$ will have a sum not far from one. In the many simulations we have done, the decreases in prediction error are almost identical whether or not the sum constraint $\sum \alpha_k = 1$ is used.

3. Stacking Trees

To see how stacking worked with trees, we set up the following program using the CART algorithms: given data $\mathcal{L} = \{(y_n, \boldsymbol{x}_n), n = 1, \ldots, N\}$, leave out some randomly selected test data $\mathcal{L}_{TS}$ resulting in $\mathcal{L}' = \mathcal{L} - \mathcal{L}_{TS}$. Divide the data in $\mathcal{L}'$ into J almost equal parts $\mathcal{L}_1, \ldots, \mathcal{L}_J$ and define $\mathcal{L}^{(j)} = \mathcal{L}' - \mathcal{L}_j$. Using the data in $\mathcal{L}^{(j)}$ grow a large tree $T_K^{(j)}$ with K terminal nodes. Prune upward so that $T_k^{(j)}$ is the subtree of $T_K^{(j)}$ having minimal learning set error among all k-node subtrees of $T_K^{(j)}$. Let $v_k^{(j)}(\boldsymbol{x})$ denote the predictor based on $T_k^{(j)}$.

Take the $\{\alpha_k\}$ to minimize

$$\sum_j \sum_{(y_n, \boldsymbol{x}_n) \in \mathcal{L}_j} (y_n - \sum_k \alpha_k v_k^{(j)}(\boldsymbol{x}_n))^2$$

under the constraints $\alpha_k \ge 0$. Grow a tree T_K based on all data in $\mathcal{L}'$, with minimum error subtrees $\{T_k\}$ and corresponding predictors $\{v_k\}$. Now estimate the error in

$$v(\boldsymbol{x}) = \sum \alpha_k v_k(\boldsymbol{x})$$

by

$$\frac{1}{N_{TS}} \sum_{(y_n, \boldsymbol{x}_n) \in \mathcal{L}_{TS}} (y_n - v(\boldsymbol{x}_n))^2 \tag{3.1}$$

where $N_{TS} = |\mathcal{L}_{TS}|$. This procedure is repeated many times with $\mathcal{L}_{TS}$ and the $\mathcal{L}_j$ selected at random and the error measures in (3.1) averaged to give an estimate of the stacking error. At the same time, the v_k having smallest cross-validated error rate is selected. Its error measure is given by

$$\frac{1}{N_{TS}} \sum_{(y_n, \boldsymbol{x}_n) \in \mathcal{L}_{TS}} (y_n - v_k(\boldsymbol{x}_n))^2$$

and this is also averaged over many repetitions.

The two data sets we experimented with were the Boston Housing data (Belsley, Kuh and Welsch (1980)) and an Ozone data set (Breiman and Friedman (1985)). The Boston Housing data has 506 cases and 12 variables. At each iteration $\mathcal{L}_{TS}$ consisted of 50 cases. One hundred iterations were run using 10-fold cross validation ($J = 10$). The Ozone data has 330 complete cases and 8 variables. Again 50 randomly selected cases were used in $\mathcal{L}_{TS}$, J set to 10, and 100 iterations. The results are shown in Table 1.

Table 1. Test Set Prediction Errors

	Data Set			
	Housing		**Ozone**	
	Best	Stacked	Best	Stacked
Error	20.9	19.0	23.9	21.6

There are other interesting results of these runs. We kept track of the average of $|\sum_k \alpha_k - 1|$ over the 100 iterations. In the housing data the result was .04 and .03 in the ozone data. The stacking combined only a small number of the predictors. In both data sets, there were usually about 50 subtrees. The average number of subtrees in a stack is 6.5 for the Boston data and 6.3 for the Ozone. The stacking ingredients are illustrated for the Housing data set. Using all data the best single tree had 23 terminal nodes. The stacking weights are given in Table 2.

Table 2. Stacking Weights

# Terminal Nodes	**Weight**
7	.29
10	.13
23	.13
26	.09
29	.12
34	.20

Suppose that the subtrees $\{T_k\}$ are nested; $T_k \subset T_{k+1}$. Then the predictor $\sum_k \alpha_k v_k(\boldsymbol{x})$ gives the same results as a single tree predictor whose size and structure is that of the largest tree in the stack. To see this, let T_m be the largest tree in the stack and t a terminal node in T_m. Define the predicted y-value $y(t)$ in t as

$$y(t) = \sum_k \alpha_k \bar{y}(t_k)$$

where t_k is the ancestor of t in T_k and $\bar{y}(t_k)$ is the average of all of the y-values falling in t_k. With this definition T_m gives the same predictions as the stacked subtrees.

The fact that the 34 terminal node tree is more accurate than the smaller 23 node tree may seem startling at first reading. But some reflection shows where the source of the increased accuracy lies. The predicted $\boldsymbol{y}$-value in each terminal node of the larger tree is based *not only* on the data in that terminal node, but also on the data in other terminal nodes with which it shares ancestors in the smaller stacked trees.

The CART pruning algorithm always produces a nested sequence of subtrees. Stacking these subtrees results in a single tree where the prediction in each terminal node "gathers strength" from the data in "nearby" terminal nodes. This result is unique to trees and no analogous interpretations are available for other types of predictors.

4. Description of the Linear Regression Simulation

Suppose the data $\{(y_n, \boldsymbol{x}_n), n = 1, \ldots, N\}$ is gotten by independent draws from a distribution $(Y, \boldsymbol{X})$ where $\boldsymbol{X} = (X_1, \ldots, X_M)$ is an M-dimensional random vector. Suppose further that this data is used to construct a predictor $\phi(\boldsymbol{x})$ of y. The prediction error of ϕ is defined as

$$PE(\phi) = E(Y^{(new)} - \phi(\boldsymbol{X}^{(new)}))^2$$

where $(Y^{(new)}, \boldsymbol{X}^{(new)})$ are random variables with the same distribution as $(Y, \boldsymbol{X})$, but independent of $\{(y_n, \boldsymbol{x}_n), n = 1, \ldots, N\}$.

Assume that

$$Y = \mu(\boldsymbol{X}) + \epsilon$$

where $E\epsilon = 0$, $E\epsilon^2 = \sigma^2$ and ϵ is independent of $\boldsymbol{X}$. Then

$$PE(\phi) = \sigma^2 + E(\mu(\boldsymbol{X}^{(new)}) - \phi(\boldsymbol{X}^{(new)}))^2.$$

The σ^2 term is the contribution of the noise in the data and is not in the experimenter's control. The second term measures how accurate an estimate ϕ is of μ and is referred to as the model error (ME) (see Breiman and Spector (1992)).

Suppose the predictor ϕ is linear; i.e. $\phi(\boldsymbol{x}) = \hat{\beta} \cdot \boldsymbol{x}$. Assume that

$$\mu(\boldsymbol{x}) = \sum_n \beta_m x_m,$$

then

$$ME = (\hat{\beta} - \beta)^t \Gamma (\hat{\beta} - \beta) \tag{4.1}$$

where Γ is the $M \times M$ covariance matrix $\Gamma_{km} = EX_kX_m$. In all of the simulations reported, the accuracy figures given will be for $N \cdot ME$ where N is the sample size (multiplying ME by N puts it on a natural scale).

In the simulations, the x-values are sampled from a 40-variable zero-mean Gaussian distribution with

$$\Gamma_{km} = R^{|k-m|}, k, m = 1, \ldots, 40.$$

The values .7, 0, -.7 are used for R.

The coefficients $\{\beta_m\}$ are in three clusters: let h be a positive integer and define

$$\begin{aligned}
&\text{if } |m-10| < h;\ \alpha_m = (h - |m-10|)^2 \\
&\text{if } |m-20| < h;\ \alpha_m = (h - |m-20|)^2 \\
&\text{if } |m-30| < h;\ \alpha_m = (h - |m-30|)^2.
\end{aligned}$$

Otherwise $\beta_m = 0$. Sample y_n from

$$Y = \beta_1 X_1 + \ldots + \beta_n X_n + \epsilon$$

where $\epsilon \in N(0,1)$, and $\beta_m = \gamma\alpha_m$. The value of γ is determined such that the theoretical R^2 given by

$$\frac{E(\sum_m \beta_m X_m)^2}{1 + E(\sum_m \beta_m X_m)^2} = .5.$$

The data-determined values of R^2 average .82 in the simulations. Five different sets of coefficients are used, corresponding to $h = 1, \ldots, 5$. For $h = 1$, there are 3 strong nonzero coefficients. For $h = 5$, there are 27 weak nonzero coefficients. The range covered includes both ends of the coefficient spectrum.

In the simulations, the data $\{(y_n, x_n), n = 1, \ldots, N\}$ were generated anew in each iteration. In all cases, 250 iterations are performed, and average ME values and their standard errors computed. There are always 40 x-variables and 60 cases.

5. Stacking Subset Regressions

In each iteration of the simulation, the variable subsets were determined using stepwise deletion. For $k = 1, \ldots, 40$ this gave a sequence of regressions such that the kth regression had k nonzero coefficients $\{\hat{\beta}_k(m)\}$, $m = 1, \ldots, 40$. To generate the level one data, for each $n = 1, \ldots, 60$, $(y_n, \boldsymbol{x}_n)$ was taken out of the data and variable deletion carried out using the remaining 59 cases. With k variables, denote the linear least squares predictor by $v_k^{(-n)}(x)$ and put

$$z_{kn} = v_k^{(-n)}(\boldsymbol{x}_n).$$

The level one data is

$$(y_n, z_{1n}, \ldots z_{40n}),\ n = 1, \ldots, 60.$$

Least squares regression constrained by coefficient nonnegativity is carried out using this data. (The algorithm and code for the constrained LS regression are given in Lawson and

Hanson (1974)). Denote the resulting coefficients by $\alpha_1, \ldots \alpha_{40}$. Then the final coefficients are given by

$$\beta(m) = \sum_1^{40} \alpha_k \hat{\beta}_k(m), \; m = 1, \ldots, 40.$$

The ME of this stacked regression is compared to the best subset regression where "best" is determined by cross-validation. For each k, $k = 1, \ldots, 40$, the estimated CV prediction error is

$$\hat{PE}(k) = \sum_{n=1}^{60} (y_n - z_{kn})^2.$$

The subset regression selected is that k minimizing $\hat{PE}(k)$. Figure 1 compares the model errors for the five coefficient sets for $R = .7, 0, -.7$. The numbers on the abscissa correspond to the coefficient sets gotten by setting $h = 1, 2, 3, 4, 5$. The triangles indicate the stacked regressions, the circles indicate the best (CV) single subset regression model error. The estimated SE's for the best single subset ME range from 1.3 to 2.0 and from .7 to 1.0 for the best stacked regression. In all cases, both procedures were used on the same data.

Although stacked regression is a clear improvement over the best subset regression, a suspicion remains that because ME is convex in the coefficients, simpler mixture methods might reduce the error as much as stacking. To see if this is true, we tried three different mixture methods. The first is an equally weighted mixture of the 5 subsets with lowest estimated CV prediction errors. The second is a similar mixture of the 10 "best" subsets.

The third is more complex. Let PE_F be the full model CV prediction error, and $\underline{PE}$ the minimum CV prediction error. Take an equally weighted mixture of all models with CV estimated PE less than $\underline{PE} + .2(PE_F - \underline{PE})$. However, if fewer than four subset models satisfy this condition, take a mixture of all models with estimated PE less than $\underline{PE} + .4(PE_F - \underline{PE})$.
The results can be summarized as:

- Stacking is uniformly best.
- Each of the three methods is sometimes close to stacking, sometimes substantially worse.

6. Stacking Ridge Regression

In ridge regression, the coefficients β are determined as the minimizers of

$$\sum_n (y_n - \boldsymbol{\beta} \cdot \boldsymbol{x}_n)^2$$

subject to the constraint $\sum_m \beta_m^2 = s$. The equivalent Lagrangian problem is to find the coefficients $\beta(\lambda)$ that minimize

$$\sum_n (y_n - \boldsymbol{\beta} \cdot \boldsymbol{x}_n)^2 + \lambda \sum_m \beta_m^2.$$

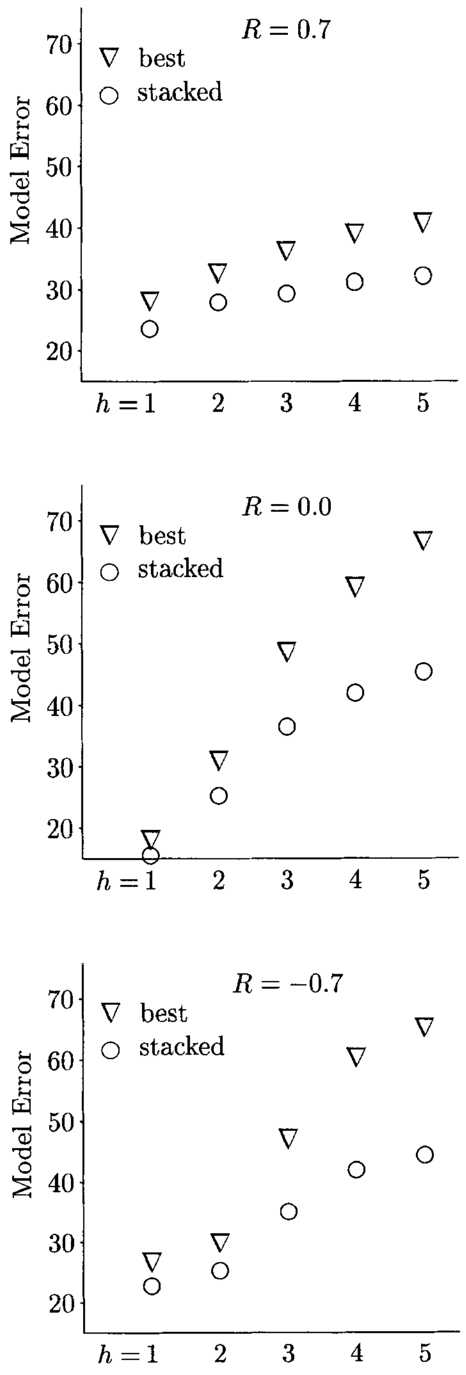

Figure 1. Model Errors for Best Subset vs Stacked Subsets.

The solution is given by

$$\beta(\lambda) = (X^t X + \lambda I) X^t y$$

where X is the data matrix. The literature gives a variety of methods for determining the value of the ridge parameter λ to be used. The method used here is cross-validation.

For each n, delete $(y_n, \boldsymbol{x}_n)$ from the data and for λ fixed compute the ridge predictor $v_\lambda^{(-n)}(\boldsymbol{x})$. The $PE(\lambda)$ cross-validation estimate is

$$\hat{PE}(\lambda) = \sum_n (y_n - v_\lambda^{(-n)}(\boldsymbol{x}_n))^2.$$

In the simulations, 40 values of λ were used; $\lambda_k = .1k^2$, $k = 1, \ldots, 40$, and the ridge regression selected corresponded to the λ_k minimizing $\hat{PE}(\lambda_k)$. The level one data was formed by setting $z_{kn} = v_{\lambda_k}^{(-n)}(\boldsymbol{x}_n)$, $k = 1, \ldots, 40$.

Figure 2 compares the model error for stacked ridge regression to the single best ridge regression. The results are sensitive to the $\boldsymbol{x}$-correlation structure and significant improvements only occur at $R = -.7$. Further discussion of this and the contrast of ridge stacking to subset stacking are deferred to Section 9. (The estimated standard errors of the MEs ranged from .4 to .9.)

7. Stacking Subset Selection With Ridge

The usefulness of stacking is most apparent when both the subset and ridge regressions are mixed together by stacking. Denote the nth case deleted k-dimensional subset predictor by $v_k^{(-n)}(\boldsymbol{x})$ and the nth case deleted ridge regression predictor using parameter λ by $v_\lambda^{(-n)}(\boldsymbol{x})$. Then, using ridge parameter values $\lambda_1, \ldots, \lambda_{K'}$, the level one data is $K + K'$ dimensional defined by

$$z_{kn} = \begin{cases} v_k^{(-n)}(\boldsymbol{x}_n), \ k \le K \\ v_{\lambda_{k-K}}^{(-n)}(\boldsymbol{x}_n), \ K < k \le K + K'. \end{cases}$$

In the next step regress y on $z_1, \ldots, z_{K+K'}$ with coefficients constrained to be nonnegative. Then use the coefficients to form a mixture of the corresponding predictors. In the simulation, $K' = 40$ and $\lambda_k = .1k^2$.

The stacking results in a predictor more accurate than either ridge or variable deletion. To provide a comparison, we use an obvious and simple combination. Select the best ridge regression by cross-validation. Similarly select the best variable subset regression. Use whichever one of this pair has the lowest estimated (CV) prediction error. Call this method "best of best". Figure 3 compares the "best of best" to stacking. Again, stacking is uniformly best, and often significantly better. (The estimated standard errors for best of best range from .7 to 2.1 and stacking from .6 to .9.)

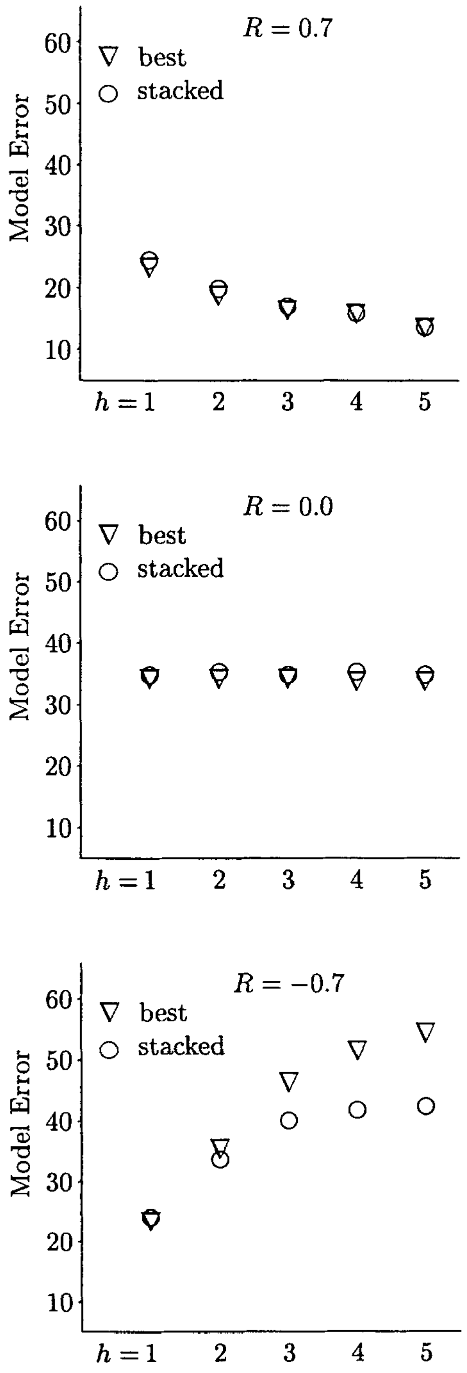

Figure 2. Model Errors for Best Ridge vs Stacked Ridge.

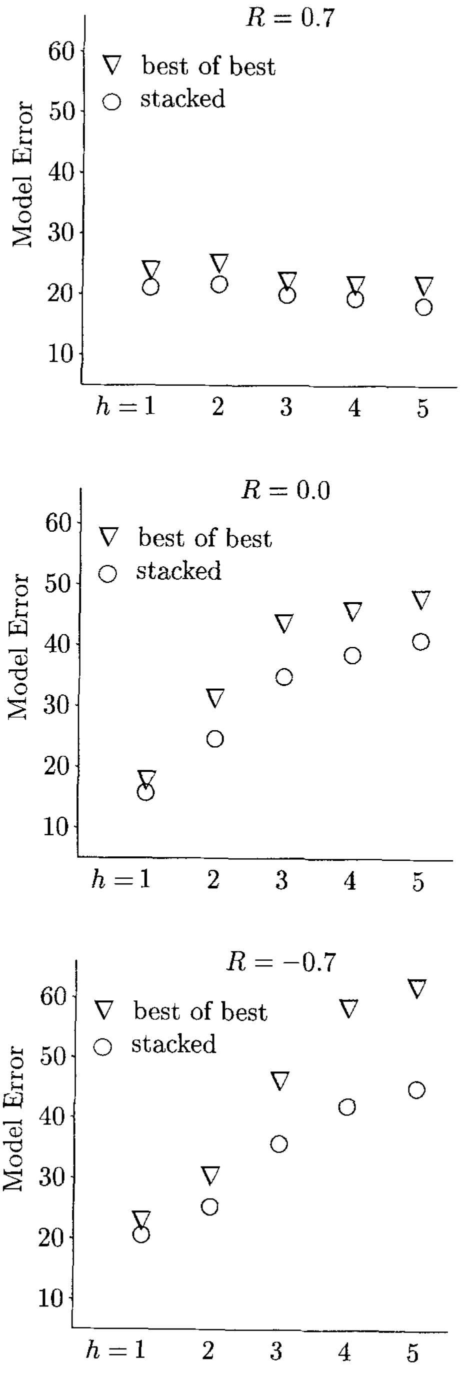

Figure 3. Model Error for "Best of Best" vs Stacked Ridge and Subset.

8. Computational Efficiency and J-Fold Cross-Validation

When the number of cases N and the number of variables are both large, generating the level one data by leave-one-out cross-validation leads to long computing runs. In this section we describe a more computationally efficient method for generating level one data and give evidence (by simulation) that it is more effective than the leave-one-out construction.

Again, suppose there are procedures for constructing linear predictors $v_1, \ldots, v_K$ using data $\{(y_n, \boldsymbol{x}_n), n = 1, \ldots, N\}$. In J-fold cross-validation (see Breiman, et. al (1984)), the data is divided at random into J-pieces $\mathcal{L}_{(1)}, \ldots, \mathcal{L}_{(J)}$ of size as nearly equal as possible. Let $\mathcal{L}^{(j)} = \mathcal{L} - \mathcal{L}_{(j)}$, where $\mathcal{L}$ is all of the data. Use $\mathcal{L}^{(j)}$ to construct the predictors $v_k^{(j)}(\boldsymbol{x})$. Then the level one data is defined by if $(y_n, \boldsymbol{x}_n) \in \mathcal{L}_{(j)}$; $z_{kn} = v_k^{(j)}(\boldsymbol{x}_n)$, $y_n = y_n$. From here on, the procedure to linearly combine the $v_1, \ldots, v_K$ is the same as before.

To compare 10-fold cross-validation stacking to leave-one-out stacking, three runs of 250 iterations each were done. The three runs corresponded to $R = .7, 0, -.7$. In the 250 iterations, the five coefficient sets were used sequentially, so that each set was used 50 times. Table 3 compares the leave-one-out MEs to the 10-fold Cv MEs.

Table 3. Comparing MEs: 10-fold CV vs Leave-One-Out CV (Number in brackets is the 10-fold ME)

	R		
	-.7	0	.7
Best Subset	45.6[43.5]	43.9[42.7]	35.3[33.8]
Best Ridge	43.3[43.5]	34.5[34.5]	17.1[17.3]
Best of Best	44.3[41.3]	36.0[32.8]	21.3[19.2]
Stacked Subsets	34.2[32.7]	33.6[31.4]	28.7[27.9]
Stacked Ridge-Subset	33.5[32.3]	31.2[29.0]	19.8[18.0]

While the differences are small, they indicate that the 10-fold data gives more accurate performance than the leave-one-out data. This is consistent with the results reported in Breiman and Spector (1992).

9. Remarks

There are interesting facets to stacked regressions. For instance, how many models are generally stacked together (i.e. combined)? That is, how many of the α_k are nonzero? The answer is–surprisingly few. For stacked subset regressions, the average number is 3.1, with the averages over the 15 simulations ($R \times h$) ranging from 2.9 to 3.4. For ridge stacking it's smaller, ranging from 1.6 to 2.3. Stacking subset and ridge together, the range is from 2.9 to 3.6 with an average of 3.4.

Another interesting question is how close is $\sum_k \alpha_k$ to 1.0. For stacked subsets the evidence is that $\sum \alpha_k$ is systematically less than 1.0 with $Av(\sum \alpha_k)$ ranging from .7 to .9 over the 15 simulations. For stacked ridge, the picture is different. For $R = .7, .0$ the range of $AV(\sum \alpha_k)$ is from 1.1 to 1.3. For $R = -.7$, the range is from .6 to 1.1. The run with $R = -.7$ and $h = 3$ was repeated using the same data as the original run but with the constraint $\sum \alpha_k = 1$ added in each of the three stacks. The ME results are given in Table 4.

Table 4. Unconstrained and Constrained MEs

	Subset	Ridge	Ridge + Subset
Original Stack	35.5	40.8	35.5
Constrained Stack	36.4	40.1	35.8

The main difference is that with the constraint more models are stacked together. Another run with $R = 0$, $h = 2$ also gave the result that the $\sum \alpha_k = 1$ constraint makes little difference in the resulting ME values.

Two general results deserve comment. First is that the results are sensitive to the value of R. In particular, whether subset or ridge regression is better depends strongly on R. This contains a lesson. Different studies in the statistical literature sometimes extoll the virtues of ridge, and sometimes of subset. All are based on simulations and the moral is that the structure of the simulation may dictate the result.

The second observation is that stacked ridge regression does not improve as much over the CV selected ridge regression as does the stacked subset regression over the CV selected subset regression. We think that the explanation is this: in subset regression, the coefficients can change considerably in going from $k + 1$ variables to k variables. However, in ridge regression the coefficients change only slightly when going from λ_k to λ_{k+1}. Therefore, mixing predictors for λ_k close to the minimizing parameter values does not produce a predictor much different from the CV minimum predictor.

Some evidence that bears on the above conjecture relates to simpler mixtures of regressions. The mixtures of subset regressions discussed in Section 5 were not as accurate as stacked regressions, but usually more accurate than the best single subset regression. Similar mixtures of ridge regressions were constructed. They were rarely more accurate than the single best ridge regression. When they were slightly better ($R = -.7$) the stacked ridge was even better.

Besides the stated simulations results, we ran hundreds of preliminary tests experimenting with ridge regression and other methods for selecting the $\{\alpha_k\}$. Some were promising, some improved on existing methods, but the non-negativity constraint method performed by far the best.

10. Conclusions

Stacking nested subtrees results in a single tree predictor with lower test set error than the best cross-validation selected subtree. Stacking various linear regressions together decreased error rates. The most improvement occurred when stacking together more dissimilar predictors. The implication is that stacking will reduce error when the predictors being stacked together are not overly similar. For instance, it seems reasonable that stacking MARS predictors based on different numbers of basis functions will give improved performance. On the other hand, if $v_k(x)$ is a neural net predictor based on K epochs of back propagation, then the similarity between the $\{v_k(x)\}$ for different k may imply that stacking will not give much improvement. Wolpert's work and its specific application in this paper should open the door to new thinking about model selection. In past statistical work, all the focus has been on selecting the "best" single model from a class of models. We may need to shift our thinking to the possibility of forming combinations of models using level one data.

References

Belsley, D.A., Kuh, E. and Welsch, R., "Regression Diagnostics," 1980, John Wiley and Sons, New York.
Berger, J.O. and Bock, M.E., "Combining independent normal mean estimation problems with unknown variances," Ann. Statist. 4, 1976, pp. 642-648.
Breiman, L., Friedman, J., Olshen, R. and Stone, J., "Classification and Regression Trees," 1984, Wadsworth, California.
Breiman, L. and Friedman, J.H., "Estimating Optimal Transformations in Multiple Regression and Correlation (with discussion)," J. Amer. Statist. Assoc., 80, 1985, pp. 580-619.
Breiman, L. and Spector, P., "Submodel Selection and Evaluation - X Random Case," International Statistical Review, 3, 1992, pp. 291-319.
Efron, B. and Morris, C., "Combining possibly related estimation problems (with discussion)," J. Roy. Statist. Soc. Ser. B, 35, 1973, pp. 379-421.
Green, E.J. and Strawderman, W.E., "A James-Stein type estimator for combining unbiased and possibly biased estimators," J. Amer. Statist. Assoc., 86, 1991, pp. 1001-1006.
Hoerl, A.E. and Kennard, R.W., "Ridge regression: Biased estimation for nonorthogonal problems," Technometrics, 12, 1970, pp. 55-67.
Lawson, J. and Hanson, R., "Solving Least Squares Problems," 1974, Prentice-Hall, New Jersey.
Luenberger, D., "Linear and Nonlinear Programming," 1984, Addison-Wesley Publishing Co.
Le Blanc, M. and Tibshirani, R., "Combining Estimates in Regression and Classification," Technical Report 9318, 1973, Dept. of Statistics, University of Toronto.
Perrone, M.P., "General Averaging Results for Convex Optimization," Proceedings of the 1993 Connectionist Models Summer School, Erlbaum Associates, 1994, pp. 364-371.
Rao, J.N.K. and Subrathmaniam, K., "Combining independent estimators and estimation in linear regression with unequal variances," Biometrics, 27, 1971, pp. 971-990.
Rubin, D.B. and Weisberg, S., "The variance of a linear combination of independent estimators using estimated weights," Biometrika, 62, 1975, pp. 708-709.
Wolpert, D., "Stacked Generalization," Neural Networks, Vol. 5, 1992, pp. 241-259.

Received July 15, 1994
Accepted October 18, 1994
Final Maunscript November 21, 1995