# The Generalization Paradox of Ensembles

John F. ELDER IV

Ensemble models—built by methods such as *bagging*, *boosting*, and *Bayesian model averaging*—appear dauntingly complex, yet tend to strongly outperform their component models on new data. Doesn't this violate "Occam's razor"—the widespread belief that "the simpler of competing alternatives is preferred"? We argue no: if complexity is measured by function rather than form—for example, according to generalized degrees of freedom (GDF)—the razor's role is restored. On a two-dimensional decision tree problem, bagging several trees is shown to actually have less GDF complexity than a single component tree, removing the generalization paradox of ensembles.

**Key Words:** Bagging; Complexity; Decision trees; Generalized degrees of freedom; Occam's razor.

## 1. MODEL ENSEMBLES

A wide variety of competing methods are available for inducing models, and their relative strengths are of keen interest. Clearly, results can depend strongly on the details of the problems addressed, as shown in Figure 1 (from Elder and Lee 1997), which plots the relative out-of-sample error of five algorithms for six public-domain problems. Every algorithm scored best or next-to-best on at least two of the six datasets. Michie, Spiegelhalter, and Taylor (1994) built a decision tree from a larger such study (23 algorithms on 22 datasets) to forecast the best algorithm to use given a dataset's properties. Though the study was skewed toward trees—they were nine of the algorithms studied and several selected datasets exhibited sharp thresholds—it did reveal some useful lessons for algorithm selection (Elder 1996a).

Still, a method for improving accuracy more powerful than tailoring the algorithm has been discovered: bundling models into ensembles. Figure 2 reveals the out-of-sample accuracy of the models of Figure 1 when they are combined four different ways, including

John F. Elder IV is Chief Scientist, Vantage Consulting Group, and Elder Research, Inc. (www.datamininglab.com), 635 Berkmar Circle, Charlottesville, VA 22901 (E-mail: elder@datamininglab.com), and Adjunct Professor, Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA.



*Journal of Computational and Graphical Statistics, Volume 12, Number 4, Pages 853–864*
*DOI: 10.1198/1061860032733*

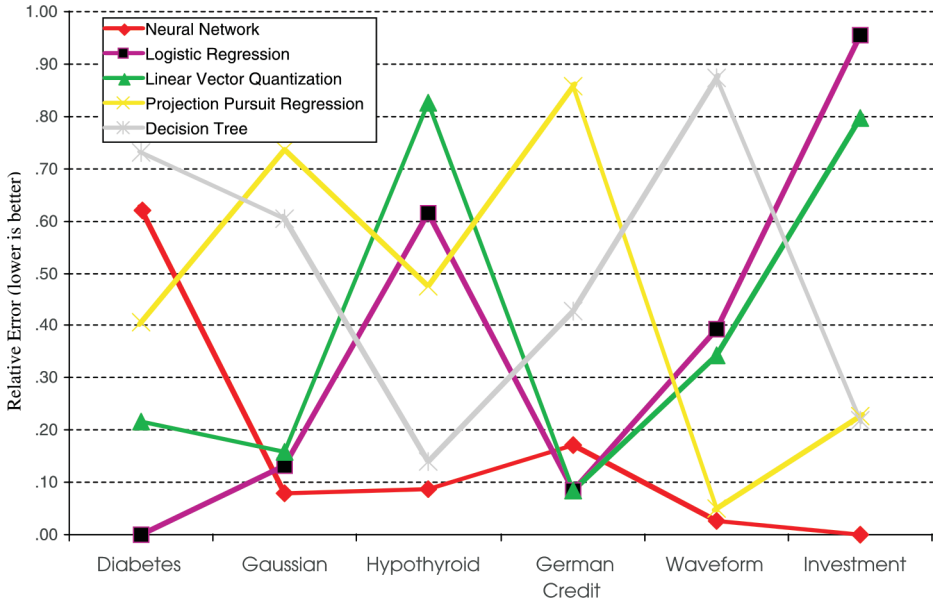


*Figure 1. Relative out-of-sample error of five algorithms on six public-domain problems (from Elder and Lee 1997).*

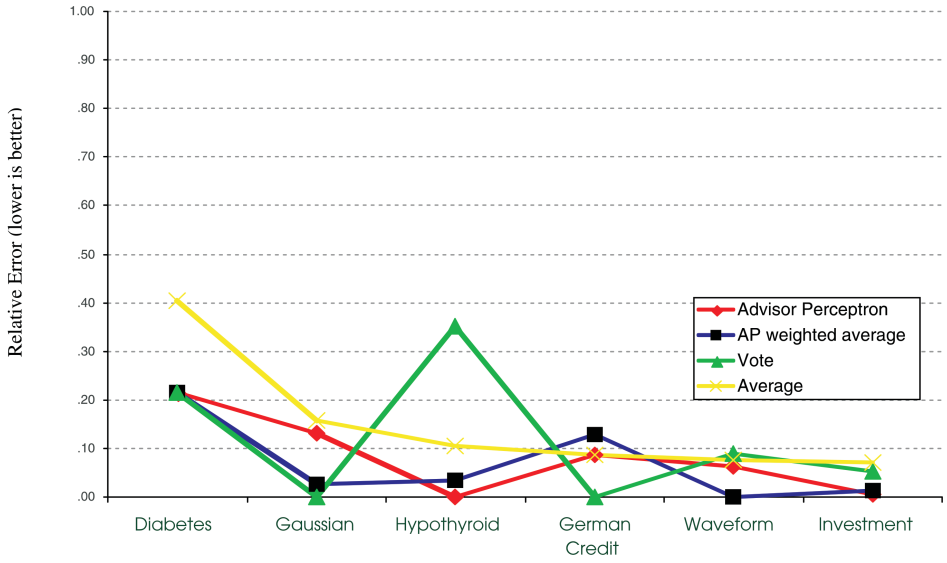


*Figure 2. Relative out-of-sample error of five ensemble methods on problems of Figure 1 (from Elder and Lee 1997).*

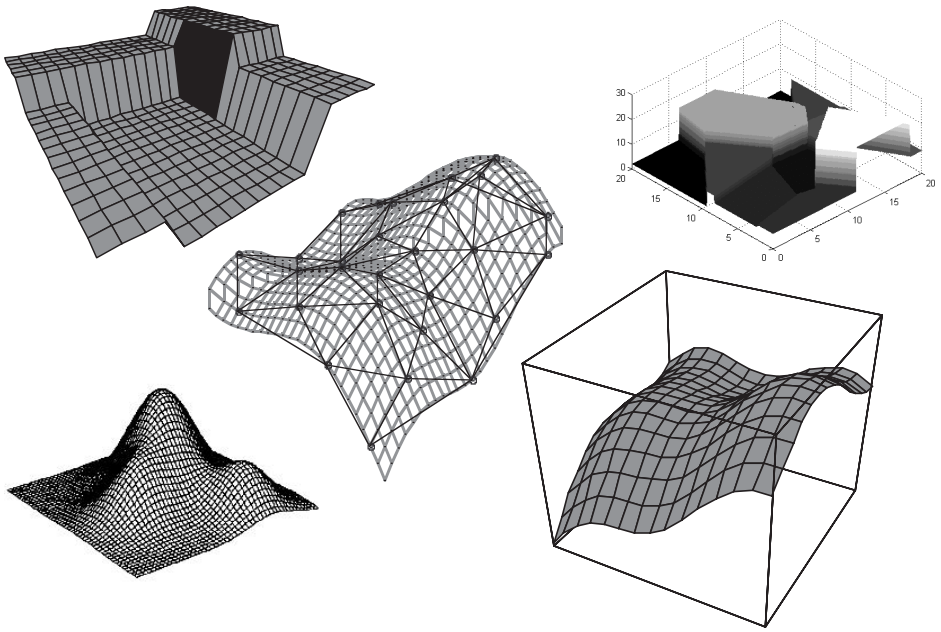


*Figure 3. Estimation surfaces of five modeling algorithms. Clockwise from top left: decision tree, nearest neighbor, polynomial network, kernel; center: Delaunay planes (Elder 1993).*

averaging, voting, and "advisor perceptrons" (Elder and Lee 1997). All four types of ensembles compare well, for each problem, with the *best* of the individual algorithms (whose identity is, of course, unknown a priori). Combining models in some reasonable manner appears more reliably accurate than trying to select the single most appropriate algorithm to employ.

Building an ensemble consists of two steps: (1) constructing varied models, and (2) combining their estimates. One may generate component models by varying case weights, data values, guidance parameters, variable subsets, or partitions of the input space. Combination can be done by voting, but is primarily accomplished through weights, with gating and advisor perceptrons as special cases. For example, Bayesian model averaging sums estimates of possible models, weighted by their posterior evidence. Bagging (*b*ootsrap *ag*gregating; Breiman 1996) bootstraps the training dataset (usually to build varied decision trees), and takes the majority vote or the average of their estimates. Boosting (Freund and Shapire 1996) and ARCing (Breiman 1996) iteratively build models by varying case weights (up-weighting cases with large current errors and down-weighting those accurately estimated) and employs the weighted sum of the estimates of the sequence of models.

The group method of data handling (GMDH; Ivakhenko 1968) and its descendent, polynomial networks (Barron et al. 1984; Elder and Brown 2000) can be thought of as early ensemble techniques. They build multiple layers of moderate-order polynomials, fit by linear regression, where variety arises from different variable sets being employed by each node. Their combination is nonlinear since the outputs of interior nodes are inputs

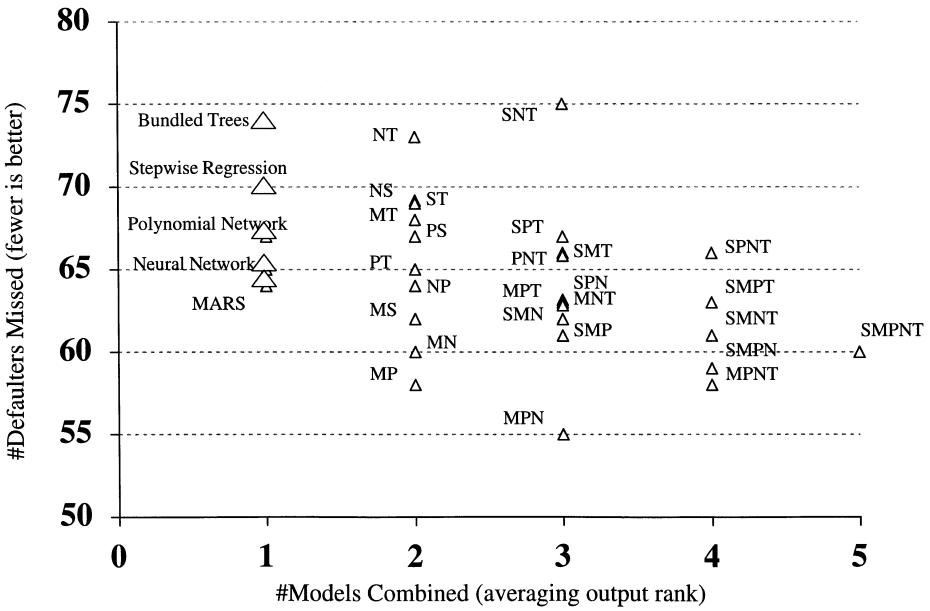


*Figure 4. Out-of-sample errors on a credit scoring application when fusing one to five different types of models into ensembles.*

to polynomial nodes in subsequent layers. Network construction is stopped by a simple cross-validation test (GMDH) or a complexity penalty. Another popular method, Stacking (Wolpert 1992) employs neural networks as components (whose variety can stem from simply using different guidance parameters, such as initialization weights), combined in a linear regression trained on leave-one-out estimates from the networks.

Finally, model fusion (Elder 1996b) achieves variety by averaging estimates of models built from very different algorithms (as in Figures 1 and 2). Their different basis functions and structures often lead to their fitting the data well in different regions, as suggested by the two-dimensional surface plots of Figure 3 for five different algorithms. Figure 4 reveals

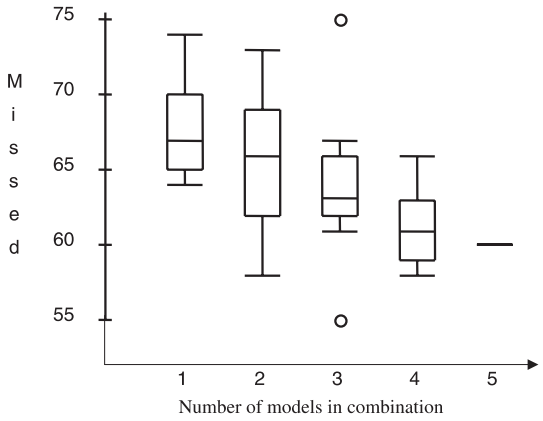


*Figure 5. Box plot for Figure 4; median (and mean) error decreased as degree of combination increased.*

the out-of-sample results of so fusing up to five different types of models on a credit scoring application. The combinations are ordered by the number of models involved, and Figure 5 highlights the finding that the mean error reduces with increasing degree of combination. Note that the final model with all five components does better than the best of the single models.

## 2. COMPLEXITY

One criticism of ensembles is that interpretation of the model is now even less possible. For example, decision trees have properties so attractive that, second to linear regression (LR), they are the modeling method most widely employed, despite having the worst accuracy of the major algorithms. Bundling trees into an ensemble makes them competitive on this crucial property, though at a serious loss in interpretability. To quantify this loss, note that an ensemble of trees can itself be represented as a tree, as it produces a piecewise constant response surface. But the tree equivalent to an ensemble can have vastly more nodes than the component trees; for example, a bag of $M$ "stumps" (single-split binary trees) can require up to $2^M$ leaves to be represented by a single tree.

Indeed, bumping (Tibshirani and Knight 1999a) was designed to get some of the benefit of bagging without requiring multiple models, in order to retain some interpretability. It builds competing models from bootstrapped datasets, and keeps only the one with least error on the original data. This typically outperforms, on new data, a model built simply on the original data, likely due to a bumped model being robust enough to do well on two related, but different datasets. But the accuracy increase is less than with ensembles.

Another criticism of ensembles—more serious to those for whom an incremental increase in accuracy is worth a multiplied decrease in interpretability—is that surely their increased complexity will lead to overfit and thus, inaccuracy on new data. In fact, not observing ensemble overfit in practical applications has helped throw into doubt, for many, the "Occam's razor" axiom that generalization is hurt by complexity. [This and other critiques of the axiom are argued in an award-winning paper by Domingues (1998).]

But, are ensembles truly complex? They appear so; but, do they *act* so? The key question is how we should measure complexity. For LR, one can merely count terms, yet this is known to fail for nonlinear models. It is possible for a single parameter in a nonlinear method to have the influence of less than a single linear parameter, or greater than several—for example, three effective degrees of freedom for each parameter in multivariate adaptive regression splines (MARS; Friedman 1991; Owen 1991). The under-linear case can occur with say, a neural network that has not trained long enough to pull all its weights into play. The over-linear case is more widely known. For example, Friedman and Silverman (1989) noted: "[The results of Hastie and Tibshirani (1985)], together with those of Hinkley (1969, 1970) and Feder (1975), indicate that the number of degrees of freedom associated with nonlinear least squares regression can be considerably more than the number of parameters involved in the fit."

The number of parameters and their degree of optimization is not all that contributes to

a model's complexity or its potential for overfit. The model form alone does not reveal the *extent of the search for structure*. For example, the winning model for the 2001 Knowledge Discovery and Data Mining (KDD) Cup employed only three variables. But the data had 140,000 candidate variables, constrained by only 2,000 cases. Given a large enough ratio of unique candidate variables to cases, searches are bound to find some variables that look explanatory even when there is no true relationship. As Hjorth (1989) warned: "... the evaluation of a selected model can not be based on that model alone, but requires information about the class of models and the selection procedure." We thus need to employ model selection metrics that include the effect of model selection!

There is a growing realization that complexity should be measured not just for a model, but for an entire modeling *procedure*, and that it is closely related to that procedure's *flexibility*. For example, the recent covariance inflation criterion (Tibshirani and Knight 1999b) fits a model and saves the estimates, then randomly shuffles the output variable, re-runs the modeling procedure, and measures the covariance between the new and old estimates. The greater the change (adaptation to randomness, or flexibility) the greater the complexity penalty needed to restrain the model from overfit. Somewhat more simply, Generalized degrees of freedom (GDF; Ye 1998) randomly perturbs (adds noise to) the output variable, re-runs the modeling procedure, and measures the changes to the estimates. Again, the more a modeling procedure adapts to match the added noise the more flexible (and therefore more complex) its model is deemed to be.

The key step in both—a randomized loop around a modeling procedure—is reminiscent of the regression analysis tool (Faraway 1991), which measured, through resampling, the robustness of results from multistep automated modeling. Whereas at that time sufficient resamples of a two-second procedure took two days, increases in computing power have made such empirical measures much more practical.

## 3. GENERALIZED DEGREES OF FREEDOM

For LR, the degrees of freedom, $K$, equal the number of terms, though this does not extrapolate to nonlinear regression. But, there exists another definition that does:

$$K = \text{trace}(\text{Hat Matrix}) = \Sigma\, \delta\hat{Y}/\delta Y, \tag{3.1}$$

where

$$\delta Y = Ye - Y, \quad \text{and} \quad \delta\hat{Y} = \widehat{Ye} - \hat{Y}, \tag{3.2}$$

$$\hat{Y} = f(Y, \mathbf{X}) \quad \text{for model } f(), \text{ output } Y, \text{ and input vectors, } \mathbf{X}; \widehat{Ye} = f(Ye, \mathbf{X}) \tag{3.3}$$

$$Ye = Y + N(0, \sigma). \tag{3.4}$$

We enjoyed naming the perturbed output, $(Y + \text{error})$ after GDF's inventor, Ye.

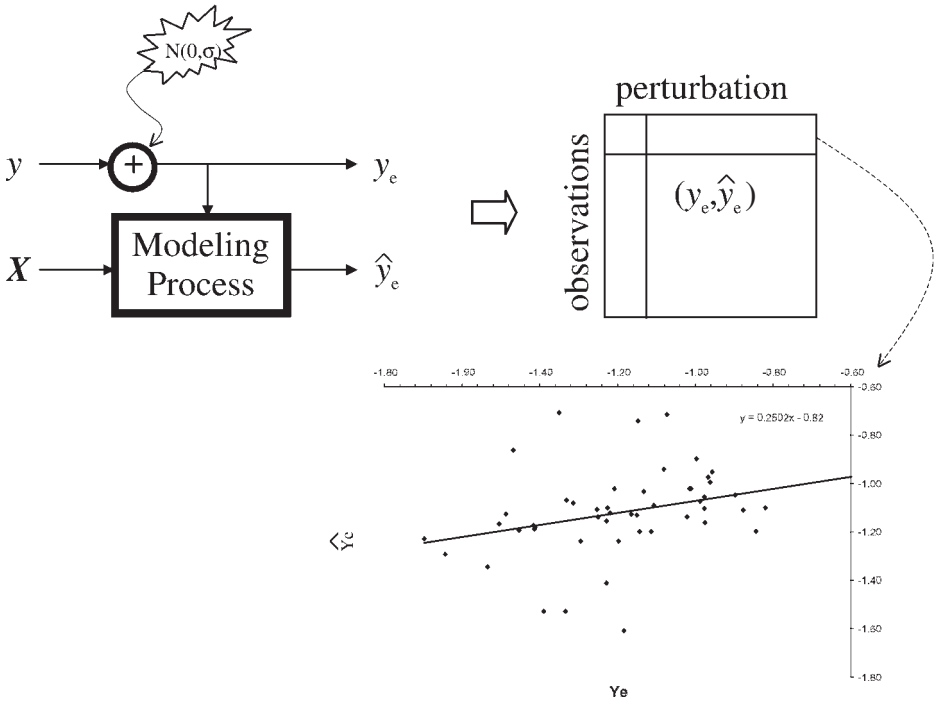


*Figure 6. Diagram of GDF computation process.*

GDF is thus defined to be the sum of the sensitivity of each fitted value, $\hat{Y}_i$, to perturbations in its corresponding output, $Y_i$. (Similarly, the effective degrees of freedom of a spline model is estimated by the trace of the projection matrix, $\mathbf{S}$: $\hat{Y} = \mathbf{S}Y$) Ye suggested generating a table of perturbation sensitivities, then employing a "horizontal" method of calculating GDF, as diagrammed in Figure 6. Fit a LR to $\delta\hat{Y}_i$ versus $\delta Y_i$ using the row of data corresponding to case $i$; then, add together the slopes, $m_i$. (Because $Y_i$ and $\hat{Y}_i$ are constant, the LR simplifies to be of $\widehat{Ye}_i$ versus $Ye_i$.) This estimate appears more robust than that obtained by the "vertical" method of averaging the value obtained for each column of data (i.e., the GDF for each model or perturbation dataset).

## 4. EXAMPLES: DECISION TREE SURFACE WITH NOISE

We take as a starting point for our tests the two-dimensional piecewise constant surface used to introduce GDF (Ye 1998), shown in Figure 7(a). It is generated by (and so can be perfectly fit by) a decision tree with five terminal (leaf) nodes (i.e., four splits), whose smallest structural change is 0.5. Figure 7(b) illustrates the "surface" after Gaussian noise $N(0, 0.5)$ has been added, and Figure 8 shows 100 random samples of that space. This tree + noise data is the $(\mathbf{X}, Y)$ dataset employed for the experiments. For GDF perturbations, we employed 50 replications, where each added to $Y$ Gaussian noise, $N(0, 0.25)$, having half the standard deviation of the noise already in the training data (a rule of thumb for

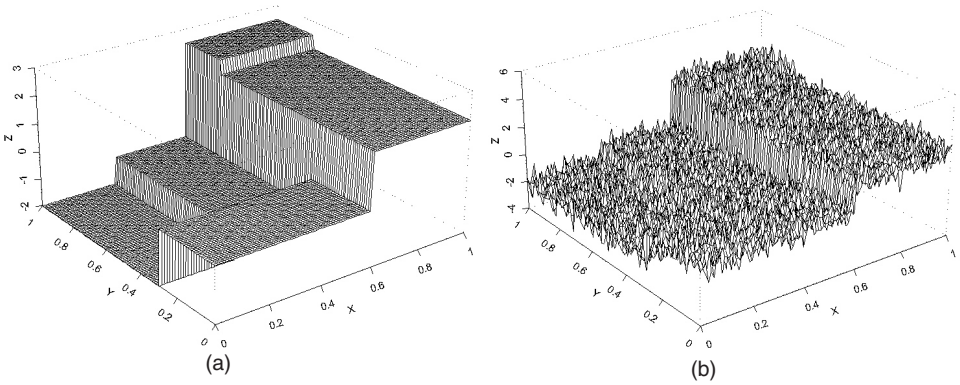


*Figure 7. (a) (Noiseless version of) two-dimensional tree surface used in experiments (after Ye 1998). (b) Tree surface of (a) after adding N(0, 0.5) noise.*

perturbation magnitude).

Figure 9 shows the GDF versus $K$ (number of parameters) sequence for LR models, single trees, and ensembles of five trees (and two more sequences described below). Confirming theory, note that the GDF for the LR models closely matches the number of terms, $K$. For decision trees of different sizes, $K$ (i.e., maximum number of split thresholds), the GDF grew at about 3.67 times the rate of $K$. Bagging (bootstrap sampling the datasets and averaging the outputs) five trees together, the rate of complexity growth is 3.05. Surprisingly perhaps, the bagged trees of a given size, $K$, are about a fifth simpler, by GDF, than each of their components!

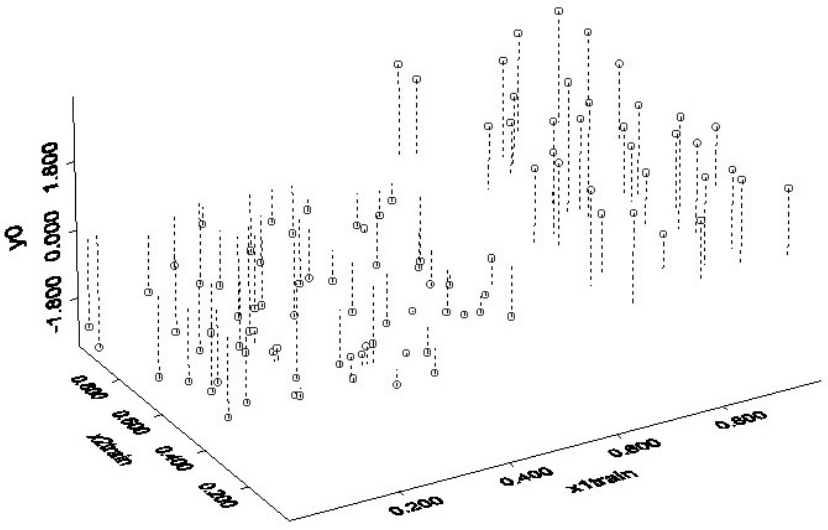


*Figure 8. 100 samples from Figure 7(b) (dotted lines connect to zero plane).*

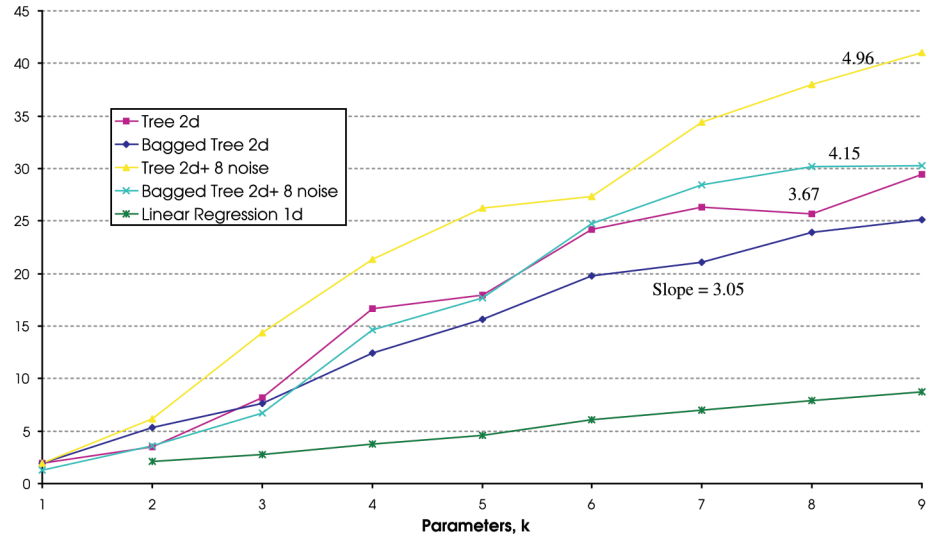


*Figure 9. GDF sequences for five models using from one to nine parameters.*

Figure 10 illustrates two of the surfaces in the sequence of bagged trees. Bagging five trees limited to four leaf nodes (three splits) each produces the estimation surface of Figure 10(a). Allowing eight leaves (seven splits) produces that of Figure 10(b). The bag of more complex trees creates a surface with finer detail (most of which here does not relate to actual structure in the underlying data-generating function, as the tree is more complex than needed). For both bags, the surface has gentler stairsteps than those of a lone tree, revealing how bagging trees can especially improve their generalization on smooth functions.

Expanding the experiment (after Ye 1998), we appended eight random candidate input variables to **X**, to introduce *selection noise*, and re-ran the sequence of individual and bagged trees. Figures 11(a) and 11(b) illustrate two of the resulting bagged surfaces (projected onto

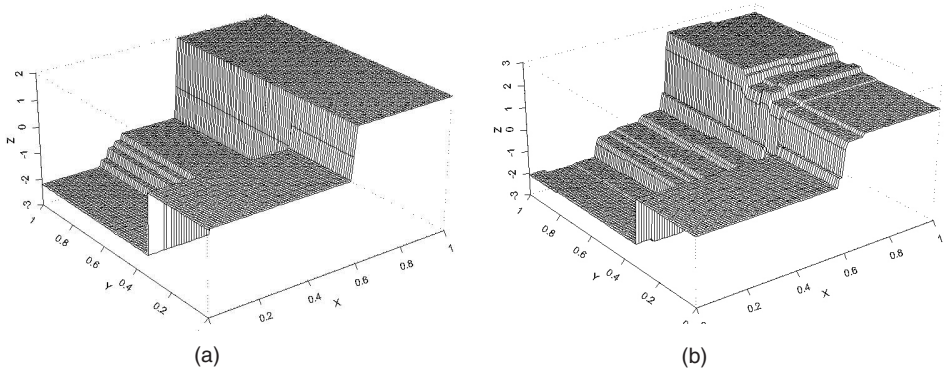


*Figure 10. Surface of bag of five trees using (a) three splits (b) seven splits.*

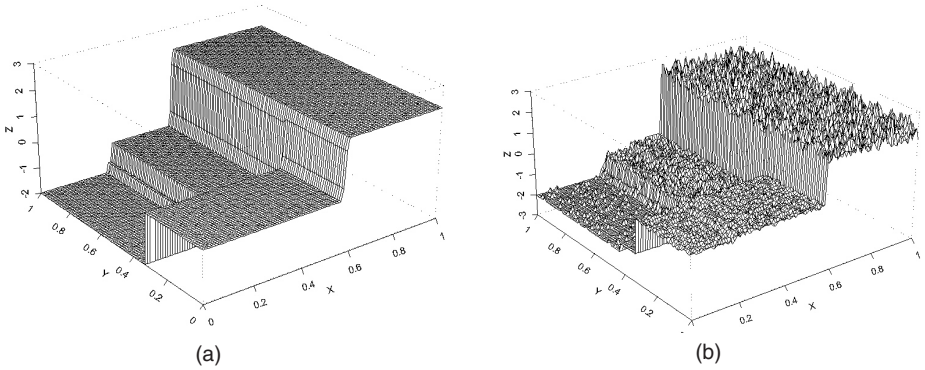


*Figure 11. (a) Surface of bag of five trees using three splits with eight noise inputs. (b) Surface of bag of five trees using seven splits with eight noise inputs (projected onto plane of two real inputs).*

the space of the two real inputs), again for component trees with three and seven splits, respectively. The structure in the data is clear enough for the under-complex model to avoid using the random inputs, but the over-complex model picks some up. The GDF progression for the individual and bagged trees with ten candidate inputs is also shown in Figure 9. Note that the complexity slope for the bag (4.15) is again less than that for its components (4.96). Note also that the complexity for each ten-input experiment is greater than its corresponding two-input one. Thus, even though one cannot tell—by looking at a final model using only the real inputs $X_1$ and $X_2$—that random variables were considered, the chance for overfit was greater, and this is appropriately reflected in the GDF measure of complexity.

## 5. SUMMARY AND DISCUSSION

Bundling competing models into ensembles almost always improves generalization—and using different algorithms is an effective way to obtain the requisite diversity of components. Ensembles appear to increase complexity, as they have many more parameters than their components; so, their ability to generalize better seems to violate the preference for simplicity embodied by "Occam's razor." Yet, if we employ GDF—an empirical measure of the *flexibility* of a modeling *process*—to measure complexity, we find that ensembles can be simpler than their components. We argue that when complexity is thereby more properly measured, Occam's razor is restored.

Under GDF, the more a modeling process can match an arbitrary change made to its output, the more complex it is. It agrees with linear theory, but can also fairly compare very different, multistage modeling processes. In our tree experiments, GDF increased in the presence of distracting input variables, and with parameter power (trees vs. LR). It is expected to also increase with search thoroughness, and to decrease with use of model priors, with parameter shrinkage, and when the structure in the data is more clear relative to the noise. Additional observations (constraints) may affect GDF either way.

Finally, case-wise (horizontal) computation of GDF has an interesting by-product:

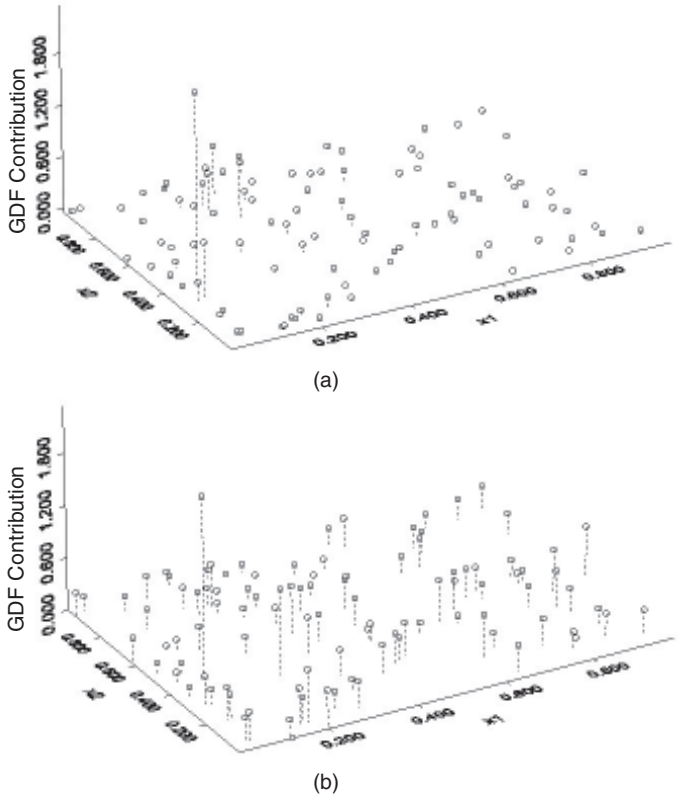


*Figure 12. Complexity contribution of each sample for bag of five trees using (a) three splits and (b) seven splits.*

an identification of the complexity contribution of each case. Figure 12 illustrates these contributions for two of the single-tree models of Figure 9 (having three and seven splits, respectively). The under-fit tree results of Figure 12(a) reveal only a few observations to be complex; that is, to lead to changes in the model's estimates when perturbed by random noise. (Contrastingly, the complexity is more diffuse for the results of the overfit tree, in Figure 12(b).) A future modeling algorithm could recursively seek such *complexity contribution outliers* and focus its attention on the local model structure necessary to reduce them, without increasing model detail in regions which are stable.

## ACKNOWLEDGMENTS

Thanks to my colleagues, Antonia de Medinacelli, Carl Hoover, and J. Dustin Hux, for helpful discussions and for programming the experiments.

*[Received November 2003. Revised December 2003.]*

## REFERENCES


Barron, R. L., Mucciardi, A. N., Cook, F. J., Craig, J. N., and Barron, A. R. (1984), "Adaptive Learning Networks: Development and Application in the United States of Algorithms Related to GMDH," in *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, ed. S.J. Farlow, New York: Marcel Dekker, pp. 25–65.

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

Domingues, P. (1998), "Occam's Two Razors: The Sharp and the Blunt," in *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, New York: AAAI Press.

Elder, IV, J. F. (1993), "Efficient Optimization through Response Surface Modeling: A GROPE Algorithm," Dissertation, School of Engineering and Applied Science, University of Virginia, Charlottesville.

——— (1996a), Review of *Machine Learning, Neural and Statistical Classification*, (eds. Michie, Spiegelhalter, and Taylor 1994), *Journal of the American Statistical Association*, 91, 436–437.

——— (1996b), "Heuristic Search for Model Structure: the Benefits of Restraining Greed," in *Learning from Data: Artificial Intelligence and Statistics*, New York: Springer-Verlag.

Elder, IV, J. F., and Brown, D. E. (2000), "Induction and Polynomial Networks," *Network Models for Control and Processing*, ed. M. D. Fraser, Portland, OR: Intellect, pp. 143–198.

Elder, IV, J. F., and Lee, S. S. (1997), "Bundling Heterogeneous Classifiers with Advisor Perceptrons," Technical Report, University of Idaho, October, 14.

Faraway, J. J. (1991), "On the Cost of Data Analysis," Technical Report, Dept. Statistics, UNC Chapel Hill.

Feder, P. I. (1975), "The Log Likelihood Ratio in Segmented Regression," *The Annals of Statistics*, 3, 84–97.

Freund, Y., and Shapire, R. E. (1996), "Experiments With a New Boosting Algorithm," *Machine Learning: Proceedings of the 13th International Conference*, July.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 19.

Friedman, J. H., and Silverman, B. W. (1989), "Flexible Parsimonious Smoothing and Additive Modeling," *Technometrics*, 31, 3–21.

Hastie, T., and Tibshirani, R. (1985), Discussion of "Projection Pursuit" by P. Huber, *The Annals of Statistics*, 13, 502–508.

Hinkley, D. V. (1969), "Inference About the Intersection in Two-Phase Regression," *Biometrika*, 56, 495–504.

——— (1970), "Inference in Two-Phase Regression," *Journal of the American Statistical Association*, 66, 736–743.

Hjorth, U. (1989), "On Model Selection in the Computer Age," *Journal of Statistical Planning and Inference*, 23, 101–115.

Ivakhenko, A. G. (1968), "The Group Method of Data Handling—A Rival of the Method of Stochastic Approximation," *Soviet Automatic Control*, 3, 43–71.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994), *Machine Learning, Neural and Statistical Classification*, New York: Ellis Horwood.

Owen, A. (1991), Discussion of "Multivariate Adaptive Regression Splines" by J. H. Friedman, *The Annals of Statistics*, 19, 82–90.

Tibshirani, R., and Knight, K. (1999a), "The Covariance Inflation Criterion for Adaptive Model Selection," *Journal of the Royal Statistical Society*, Series B, 61, 529–546.

——— (1999b), "Model Search and Inference by Bootstrap 'Bumping,' " *Journal of Computational and Graphical Statistics*, 8, 671–686.

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

Ye, J. (1998), "On Measuring and Correcting the Effects of Data Mining and Model Selection," *Journal of the American Statistical Association*, 93, 120–131.