Visualizing Bagged Decision Trees

J. Sunil Rao
Department of Biostatistics
Cleveland Clinic, srao@bio.ri.ccf.org

William J.E. Potts
Professional Services Division
SAS Institute Inc, saswzp@wnt.sas.com

Abstract

We present a visual tablet for exploring the nature of a bagged decision tree (Breiman [1996]). Aggregating classifiers over bootstrap datasets (bagging) can result in greatly improved prediction accuracy. Bagging is motivated as a variance reduction technique, but it is considered a black box with respect to interpretation. Current research seeking to explain why bagging works has focused on different bias/variance decompositions of prediction error. We show that bagging's complexity can be better understood by a simple graphical technique that allows visualizing the bagged decision boundary in low-dimensional situations. We then show that bagging can be heuristically motivated as a method to enhance local adaptivity of the boundary. Some simulated examples are presented to illustrate the technique.

Introduction

Decision trees are flexible classifiers with simple and interpretable structures (Ripley [1996]). The best known methods for constructing decision trees are CART (Breiman *et al.* [1984]) and C4.5 (Quinlan, [1993]). Consider a learning sample $\mathcal{L}$ consisting of a $p-$vector of input variables and a class label for each of n cases. Tree-structured classifiers recursively partition the input space into rectangular regions with different class assignments. The resulting partition can be represented as a simple decision tree. These models are, however, unstable to small perturbations in the learning samples — that is, diffferent data can give very different looking trees.

Breiman [1996a] introduced *bagging* (bootstrap aggregation) as a method to enhance the accuracy of unstable classification methods like decision trees. In bagging, B bootstrap (Efron and Tibshirani [1993]) datasets, are generated, each consisting of n cases drawn at random but with replacement from $\mathcal{L}$. A decision tree is built for each of the B samples. The predicted class corresponding to a new input is obtained by a plurality vote among the B classifiers.

Consequently, each new case must be run down each of the B decision trees and a running tally kept of the results. Bagging decision trees has been shown to lead to consistent improvements in prediction accuracy (Breiman [1996a,b], Quinlan [1996]).

Bagging takes advantage of instability to improve the accuracy of the classification rule, but in the process destroys the simple interpretation of a single decision tree. Bagging stable classifiers can however actually *increase* prediction error (Breiman [1996a]). A flurry of current work to understand the theoretical nature of bagging has focused on different bias/variance decompositions of prediction error (Brieman [1996a,b], Friedman [1996], Tibshirani [1996], Kohavi and Wolpert [1996], James and Hastie[1997]). For simple risk functions like squared error loss, bagging can be shown to improve prediction accuracy through variance reduction. But due to the non-convexity of a $0-1$ misclassification rate loss function, there is not a simple additive breakdown of prediction error into bias plus variance. What has been shown is that there is an interesting interaction between (boundary) bias (the decision rule produced relative to the gold standard Bayes rule) and variance of the classifier, and that depending on the magnitude and sign of the bias, bagging can help or do harm.

Leaving aside the algebraic decompositions, bagging is generally regarded as a black box — it's inner workings cannot be easily visualized or interpreted. In this paper, we use a new graphical display called a *classification aggregation tablet* scan or CAT scan to visualize the bagging process for low dimensional problems. This is a general graphic that can be applied to any aggregated classifier. Here however, we focus on decision trees for the two-class discrimination problem with two-dimensional input vectors.

The CAT Scan

For each learning sample $\mathcal{L}$ and set of B bootstrap decision trees, a single CAT scan can be produced. The CAT scan was constructed using a *small multiple* design (Tufte [1990]) in order to effectively display the cumulative effect of bagging. Each CAT scan consists of a two-dimensional array of images. The coordinate system of each individual image represents the two-dimensional input space.

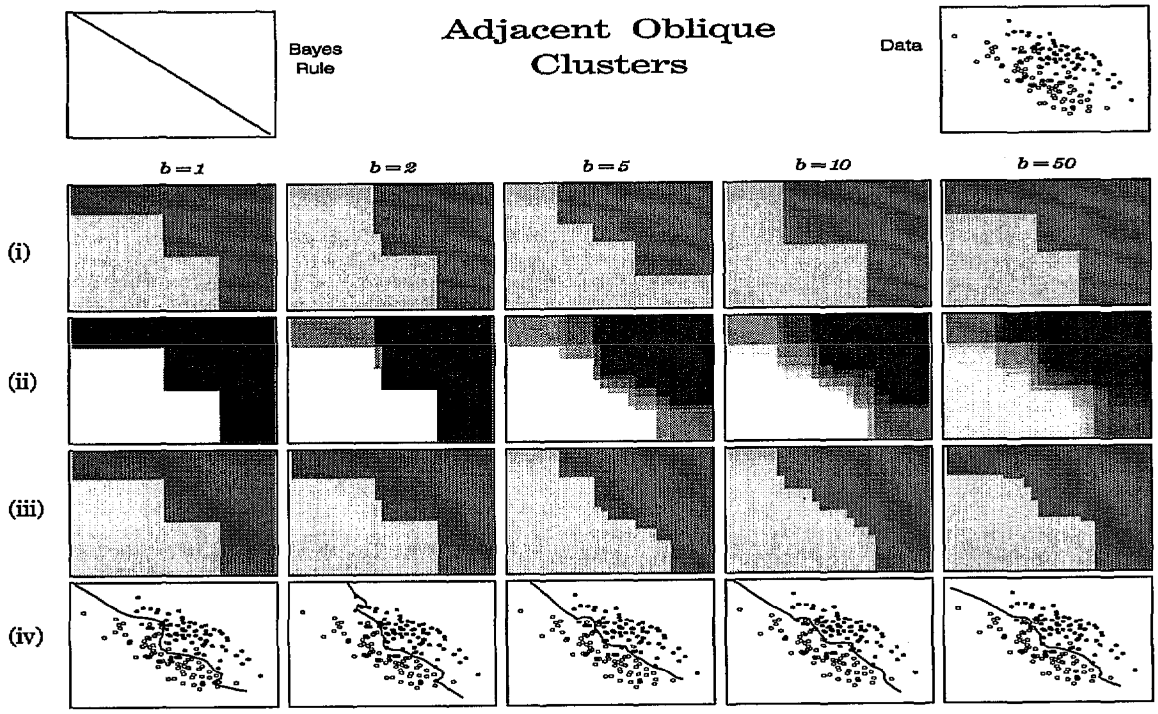

Figure 1: CAT scan for adjacent oblique clusters example.

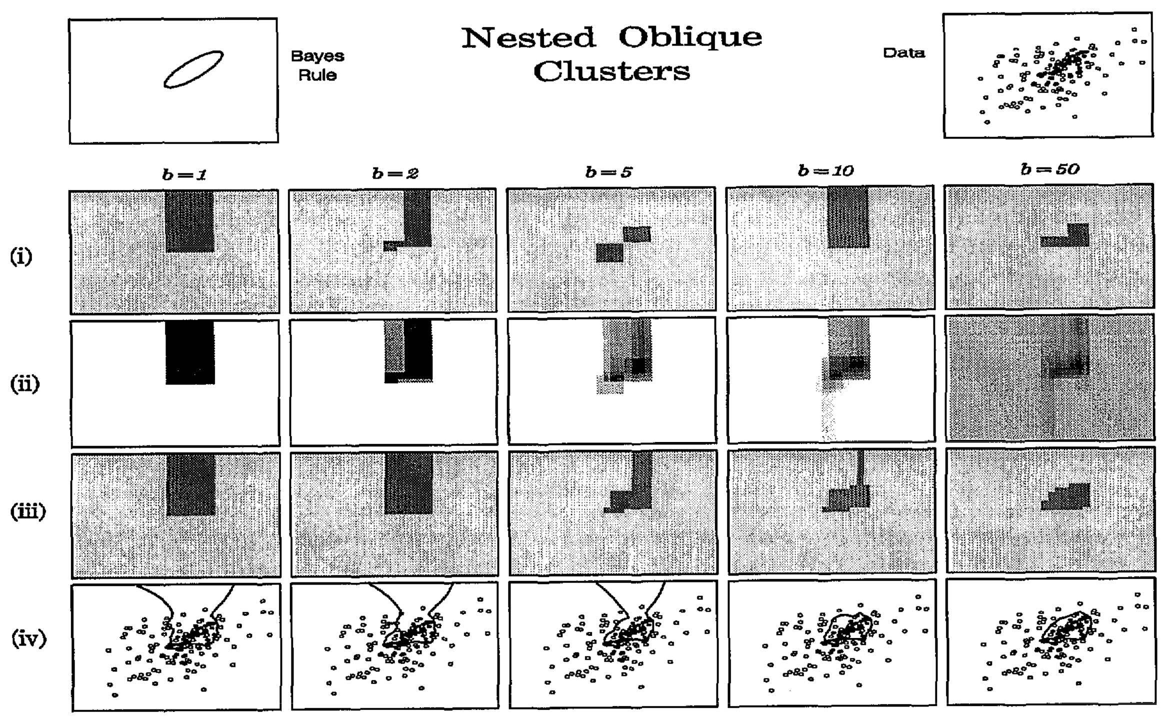

Figure 2: CAT scan for nested oblique clusters example.

aggregating classifiers. Bootstrap aggregation is just one special case. Although presented for the two-dimensional case, the CAT scan can potentially be generalized to higher dimensions using a grand tour style approach. This would involve significantly more computation and while theoretically feasible, was not the main intent of this paper. We sought simply to visualize the smoothing of a decision boundary by bagging and hence focused on low dimensional views and the two-class problem. We have only explored bagged decision trees, but the CAT scan could be used to examine other methods where bagging can be potentially detrimental, such as nearest neighbour classifiers (Breiman [1996a]).

What is clear from the studying the simulated examples, is that bagging is not a black box. It can be thought of a member of the general class of flexible discriminants (Ripley [1996]). It gives a flexible decision boundary with the ability to effectively model oblique and nonlinear Bayes rules.

The decision region for a classification tree is rectilinear with segments parallel to the input axes. This boundary defines a decision region for each class. A bagged decision tree is the union of the intersection of many of these regions. For example, if $B = 3$ and the decision regions for the three trees are R_1, R_2, and R_3, then the bagged decision region would be $(R_1 \cap R_2) \cup (R_1 \cap R_3) \cup (R_2 \cap R_3)$. So that, the bagged decision region also has a rectilinear boundary composed of axis-parallel segments, as the CAT scans clearly show. In principle, a single decision tree could give the same decision boundary; but, in practice, they do not.

So why can't a single tree find the same decision boundary as bagging? To answer this question one needs to explore how the respective boundaries differ. The obvious difference, apparent on the CAT scans, is that the resolution of the bagged boundary is much finer. That is, the bagged boundary is composed of much smaller segments and thus can capture finer detail. The main reason, in practice, that a single decision tree does not give a boundary with this fine resolution is that they run out of data. For a single tree, the small segments would have to correspond to partitions of subsets of the data. But many of these small segments would correspond to partitions of subsets with little or no data in them. In contrast, bagging constructs these small segments by the union of the intersection of many larger partitions and thus does not have this problem.

Even with enough data to make the necessary splits, a single decision tree could not duplicate the bagged decision boundary without a large increase in variance. The size of a decision tree is controlled by the pruning of a large (maximal) tree. Pruning reduces variance and increases accuracy (Breiman *et al.* [1984]). If pruning was to accomodate fine splits of the data in some regions of the input space, it would also accomodate fine splits in other regions. The decision regions of such trees would not appear as large homogeneous blocks as in row (i) of the CAT scans. They would appear more like a checker board pattern. Bagging is able to locally adapt (and smooth) the decision boundary to regions of the input space that require more or less complexity.

References

1. Breiman, L. [1996a] Bagging predictors, *Machine Learning*, 24, 123-140.
2. Breiman, L. [1996b] Bias, variance, and arcing classifiers, Technical Report 460, Department of Statistics, University of California, Berkeley.
3. Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. [1984] *Classification and Regression Trees*, Wadsworth Publishers.
4. Efron, B. and Tibshirani, R.J. [1993] *An Introduction to the Bootstrap*, Chapman and Hall.
5. Friedman, J. [1996] On bias, variance, 0/1 loss, and the curse of dimensionality. Technical Report, Department of Statistics, Stanford University.
6. James, G. and Hastie, T. [1997] Generalizations of the bias/variance decomposition for prediction error. Technical Report, Department of Statistics, Stanford University.
7. Kohavi, R. and Wolpert, D. [1996] Bias plus variance decomposition for zero-one loss functions. *Machine Learnig: Proceedings of the Thirteenth International Conference* (to appear).
8. Quinlan, J.R. [1993] *C4.5: Programs for Machine Learning*, Morgan Kaufmann.
9. Quinlan, J.R. [1996] Bagging, boosting and C4.5. *Proceedings of the 13th American Association for Artificial Intelligence National Conference*, 725-730, AII Press.
10. Ripley, B.D. [1994] Neural networks and related methods for classification. J. R. Statist. Soc. B, 56, 409-456.
11. Ripley, B.D. [1996] *Pattern Recognition and Neural Networks*, Cambridge University Press.
12. Tibshirani, R. [1996] Bias, variance and prediction error for classification rules, Technical Report, Department of Statistics, University of Toronto.
13. Tufte, E. R. [1990] *Envisioning Information*, Graphics Press, Cheshire, Conn.