Managing Multiple Models

Hugh A. Chipman
Department of Statistics
and Actuarial Science
University of Waterloo
Waterloo, Ontario, N2L 3G1
hachipman@uwaterloo.ca

Edward I. George
Department of Management Science
and Information Systems
University of Texas
Austin, TX 78712-1175
egeorge@mail.utexas.edu

Robert E. McCulloch
Graduate School of Business
University of Chicago
Chicago IL, 60637
robert.mcculloch@gsb.uchicago.edu

Abstract

Recent research in model selection and adaptive modeling has produced an embarrassment of riches. By using any one of several different techniques, an analyst is able to generate a number of models that describe the same data set well. Examples include multiple tree models generated by bootstrapping or stochastic searches, and different subsets of variables in linear regression models identified by stochastic or exhaustive searches. While model averaging can use these models to improve prediction accuracy, interpretation of the resultant models becomes difficult. We seek a compromise, developing measures of dissimilarity between different models and using these to select good models which may reveal different aspects of the data. Data on housing prices in Boston are used to illustrate this in the context of treed regression models.

1 INTRODUCTION

The problem of model uncertainty occurs in many data analyses. Having specified a family of models (for example linear regression models), a variety of submodels from this family may describe the data well (in the linear regression case, for example, different subsets of variables). Two common approaches to the model uncertainty problem are to either choose a single model, or average predictions across models. In the latter case, methods such as Bagging (Breiman 1996), Boosting (Freund and Schapire 1996), or Bayesian model averaging can be used.

This paper introduces techniques for the exploration of the set of plausible models. These techniques are based on the notion of dissimilarities between pairs of models. We will argue an understanding of the distribution of models is useful, whether selection or averaging is the goal. If a single model is to be selected, we will be more reassured if all other plausible models are quite similar to this model. If models are to be averaged over, better predictions may result if we ensure that different models are included in the average predictions. Breiman (1999) has observed this property in the context of forests of trees. A compromise between selection and averaging may be possible, by selecting a handful of "representative" models, and averaging over them, rather than averaging over all models. Provided that this set of models is small enough, there is still the possibility of interpreting them as one would a single model.

We focus here on methods for assessing differences between models, and selecting and interpreting representative models. This approach is predicated on the notion that interpreting one or more models is of interest. If a "black box" predictor that minimizes out-of-sample prediction error is desired, methods for managing multiple models will not offer an improvement over model averaging.

To assess differences between models, metrics both within a class of models and between different classes of models are developed, and used to cluster models into similar groups. Although many of the ideas are general, the focus here will be on linear models and trees, with examples using tree models.

To illustrate the problem of multiple models, consider the Boston Housing data (Harrison and Rubinfeld 1978). 506 census tracts in the greater Boston area have 14 recorded characteristics, such as crime rate, proportion of lower status individuals, parent-teacher ratios, pollution levels, median house value, and others. The goal is to predict median house value using the other 13 variables as predictors. The data have been used to illustrate regression diagnostics (Belsley, Kuh and Welsch 1980), and there are strong indications of local behaviour in the data. Further discussion of the data is given in section 4.

```
nox                         rad
 | * n=232                    | ptratio
 | chas                       |    | * n=204
     | lstat                  |    | * n=54
     |    | nox               | dis
     |    |    | * n=47            | * n=27
     |    |    | * n=34            | * n=52
     |    | * n=15
     | * n=9
```

Figure 1: Two treed regression models identified by Bayesian CART. Each line corresponds to a node. Parent nodes list the splitting variable. Terminal nodes are indicated with a *, and have sample size of the training dataset indicated. The vertical lines and indentations indicate the two children of each parent node. The eight variables appearing are nitrogen oxides concentrations, Charles river indicator, percent of population lower status, radial highway accessibility index, pupil-teacher ratio, and distance to employment centers.

A treed regression (Chipman, George and McCulloch 2000) model was applied to this data. In this model, a tree is constructed using a Bayesian stochastic search algorithm. This tree differs from conventional trees in that each terminal node contains a linear model. One feature of the Bayesian estimation procedure is that multiple trees are generated. In Figure 1, two such trees are illustrated. Both trees fit the data very well, but use completely different variables to partition the predictor space into regions where linear models are applied.

Thousands of other distinct treed regressions were visited by our stochastic search procedure. Many of those fit the data well (just 23 were kept for analysis presented later in this paper). These two trees and others like them suggest key questions: How different are the good models? How can we decide which models are worth examining? How many models should we look at?

Until recently, common practice has been to produce tree diagrams like those in Figure 1 for a number of trees with good fit, and select trees by hand. A more automatic and quantitative approach is proposed here, in which models are clustered according to several metrics.

In Section 2, several methods for producing multiple models are reviewed. Section 3 discusses several distance measures for models, and in Section 4 an example is given to illustrate how these measures may be used to select trees and other models. Related work and other problems are discussed in Section 5.

2 METHODS FOR GENERATING MULTIPLE MODELS

Although the focus of the paper is trees and linear models, many methods for generating multiple models are more broadly applicable. They are presented in this broader context, with specific references to trees or linear models as required.

Finding multiple models is a challenging problem, and one that is not satisfactorily addressed by the "greedy" stepwise algorithm. This algorithm makes additions or deletions from the current model in small steps (adding or deleting a single variable in linear models, or growing or pruning a node in trees). The model selected by greedy algorithms is only locally optimal. If multiple distinct models exist, identification of one good model is the best one can hope for with the greedy algorithm.

In very small problems, or those with simple structure, exhaustive searches over the model space are an effective alternative to greedy local searches. For example, in linear regression with small to moderate numbers of predictors (say less than 40), branch-and-bound algorithms (Furnival and Wilson 1974) are feasible.

Other algorithms can be used which are faster than an exhaustive search and more complete than a greedy search. They often involve either manipulation of the training data or modification of the search method. Breiman (1996) and Tibshirani and Knight (1999) propose random manipulation of the training data via the bootstrap (called "bagging" and "bumping" respectively). By perturbing the data, the greedy search identifies different models, some of which may be close to a global or local maxima.

Freund and Schapire (1996) propose an algorithm (called "boosting") for generating and combining a sequence of classification models. The data are iteratively reweighted instead of randomly resampled. The weights are adaptively chosen, with more weight given to observations that are predicted poorly. Again, multiple models result.

The second group of algorithms introduce a stochastic element to the search rather than manipulating the data. Bayesian methods that use stochastic searches via Markov chain Monte Carlo (MCMC) have been implemented for both linear models (George and McCulloch (1993)), trees (Chipman et. al. (1998a), Denison, Mallick and Smith (1998)) and treed regressions (Chipman, George and McCulloch 2000). As with the greedy algorithm, the space of models is traversed by small steps. An important difference from the greedy algorithm is the stochastic choice of a step, rather than always selecting the best local modification to the model.

Other algorithms for stochastically constructing models have been recently developed, especially in the context of trees. These include simulated annealing (Lutsko and Kuijpers 1994) and randomized greedy methods (an overview is provided in Breiman 1999).

3 METRICS ON MODELS

Given a set of models, a general approach toward organizing them is to treat each model as a point in a complex high-dimensional space. These models could then be clustered according to some dissimilarity or distance measure. Somewhat informally, we refer to these distances as metrics even if the triangle inequality does not hold. Obviously, this space is much richer and more complicated than Euclidean space, and will depend on the class of models under consideration. To facilitate development of metrics, note that a tree and a linear model can be identified by a finite set of parameters, and these parameters can be broadly divided in two groups: a structural component (the subset of variables in a linear model, or the tree structure and split rules in nodes) and a continuous component (the regression coefficients for the variables included in a linear model, and the parametric models in each terminal node). Metrics may be defined on either the structural or the continuous components, or perhaps both.

With linear models, a natural metric can be defined by looking at the subset of variables included in the model as a binary vector γ. The *matching coefficient*, one of the most common dissimilarities for categorical data, can be applied to γ. The matching coefficient is the number (or percentage) of elements which are different. So with 10 predictors, the models $M_1 = \{A\}$ with $\gamma_1 = (1,0,0,0,0,0,0,0,0,0)$ and $M_2 = \{A,B,C\}$ with $\gamma_2 = (1,1,1,0,0,0,0,0,0,0)$ would have a distance of 2 (or 20%) since both contain A, and B and C appear only in M_2. This metric has an interesting graphical property. If a graph is constructed with vertices corresponding to models and edges linking models that differ by a distance of 1 (not a percentage), then the distance between any two models corresponds to the shortest path between corresponding vertices in the graph. Another dissimilarity measure might be asymmetric, such as the *Jaccard coefficient*, which discards any 0-0 matches. In the above example, there are only 3 variables active in the two models, and they agree on a single variable (namely A). This gives a dissimilarity of $2/3 = 67\%$.

The Jaccard and matching coefficients do not capture the sign or magnitude of the regression coefficients. For example, in the Boston data, if two separate regressions are fit to the tracts in the city and those in the suburbs, the sign of the coefficient for the average number of rooms is negative for the city and positive for the suburbs. Increasing room size in the city must be a surrogate for some other undesirable (and unobserved) characteristic, while in the suburbs, big houses just cost more. It seems intuitive that these two models should be more different than if the coefficient was positive in one model and zero in the other. A metric that takes this into account would multiply the binary vector γ by the sign of the corresponding regression coefficient. This would mean that if a coefficient changes sign from one model to another, the models are twice as different as if the coefficient was important in one model and unimportant in the other. A dissimilarity measure might also be applied to the coefficient vector β for each model, but this is not explored here.

Another metric would be to compare the predictions of the two models. Let M_1, M_2 be two models. They have been trained using the same n observations $(y_i, \mathbf{x}_i), i = 1, \ldots, n$. For each observation $y_i, i = 1, \ldots n$ we have an associated *fitted value* $\hat{y}_{ij}$ for model j. The fit metric would be given by

$$d(M_1, M_2) = \frac{1}{n} \sum_{i=1}^{n} m(\hat{y}_{i1}, \hat{y}_{i2}), \tag{1}$$

where m is a metric on the fitted values. For regression models with a continuous response, natural choices would be

$$m(y_1, y_2) = (y_1 - y_2)^2 \tag{2}$$

or $m(y_1, y_2) = |y_1 - y_2|$. For classification models, $\hat{y}_{ij}$ might be the estimated class for observation y_i, in which case we could compare classifications by

$$m(y_1, y_2) = \begin{cases} 1 & \text{if } y_1 \neq y_2 \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Metrics on the estimated class probabilities $(\hat{p}_{1j}, \ldots, \hat{p}_{cj})$, $j = 1, 2$ (for c response classes) are also possible.

The fit metric above is evaluated on the training set $y_1, \ldots y_n$. It should better discriminate between models if evaluated on a test set, since in the training set, all models attempt to come as close as possible to the observed responses. In this paper, test data only will be used to construct the fit metric.

Any model that produces predictions can be compared with the fit metric. This makes it possible to compute a dissimilarity between models in the same class, across different classes, or a dissimilarity between a model and the observed y_i's. This is explored in Section 4.

We now consider several metrics applicable specifically to tree models. Because trees have a richer structure, a wider variety of metrics are possible. Additional details of these metrics are provided in Chipman et. al. (1998b).

Rather than using the fitted values, a metric could be defined on the manner in which trees partition the predictor space. Trees which are very similar will place the same observations together and separate the same observations. Andrews (personal communication) suggests the following metric. Let $I_1(i,k)$ be 1 if T_1 places observations i and k in the same node and 0 otherwise. For a *partition metric*, we look at differences between I for the two trees:

$$d(T_1, T_2) = \frac{\sum_{i>k} |I_1(i,k) - I_2(i,k)|}{\binom{n}{2}}. \qquad (4)$$

The factor $\binom{n}{2}$ scales the metric to the range (0,1) with 0 indicating perfect agreement. This metric can be interpreted as the percentage of all pairs of observations that are assigned to the same terminal node. It can be calculated efficiently using a frequency table of the terminal node labels for each data point from the two trees. As with the fit metric, test data seems more likely to discriminate models than training data.

The fit and partition metrics do not utilize the topology of the tree - they only use the observed responses and the partition defined by the terminal nodes. Shannon and Banks (1998) propose a *tree metric* which accounts for the manner in which the tree is constructed. This metric compares rules at nodes in the same position in the two trees. That is, if two plots are constructed on transparent paper so that nodes in the same position overlap and the plots are held up to the light, the metric counts the number of nodes at which the splitting rules are discrepant. The distance between trees is then a weighted sum of the discrepancies at each location:

$$d(T_1, T_2) = \sum_{r \in \text{nodes}(T_1,T_2)} \alpha_r m(\text{rule}(T_1, r), \text{rule}(T_2, r)) \quad (5)$$

The summation is over all node positions r which are nonterminal in at least one tree. The metric m compares the rules at two nodes; Shannon and Banks take m to be 1 whenever the same variable is used (no matter what splitting rule is used within a variable), and 0 otherwise. Choosing all weights $\alpha_r = 1$ yields a count of the number of nodes at which the rules differ. This metric can capture tree structure, but two isomorphic trees with the same splits in a different order will be identified as dissimilar.

The tree models considered in this paper are "treed regressions" which have a linear model in each terminal node. The partition and tree metrics are equally effective on conventional trees, which have simple means or proportions in a terminal node. Caution should be exercised in computing metrics between different classes of trees, such as a conventional tree and a treed regression. Conventional trees partition the predictor space so that constant models apply well in terminal nodes, while treed regressions seek partitions where linear models apply. The partitions and the trees used to generate them need not be comparable.

4 AN EXAMPLE

In this section we discuss several techniques for managing multiple models, using the Boston housing data. Another example using breast cancer data is discussed in Chipman et. al. (1998b).

A training set of 337 out of 506 observations (2/3 of the data) was selected randomly. Following previous analyses of this data, the log of median house value was taken to be the response. Both the response and predictors are rescaled so that each has mean zero and range 1.

Treed regression models are constructed using the Bayesian procedure described in Chipman, George and McCulloch (2000). Ten runs of the Metropolis Hastings chain were used to search the tree space, with 1000 steps being taken within each chain. In each of these ten runs, the best trees of each size visited were saved. All chains produced trees of sizes 2,3,4,5 and some of size 6 were also produced. Automatic choices of prior parameters discussed in Chipman et. al. (2000) were used, setting $c = 3$, giving a sufficiently broad spread to the prior on regression coefficients β. We chose a prior on tree size with an average size of about 2 terminal nodes and a prior on the residual noise variance such that 75% of probability is on smaller variances than in a single linear regression model.

The best trees of each size from the 10 runs were further screened to remove duplicates and trees that fit poorly. Duplicates were defined to be those trees with an identical log integrated likelihood (hereafter called log likelihood, see Chipman et. al. 1998a for details). Poor fit was indicated by a log likelihood that was more than 20 below the largest log likelihood. This yielded a set of 23 trees. Interestingly, one of the 23 trees corresponds to a partition of the data into city and suburb, which can be verified using names of the 506 census tracts.

For each of the 23 trees, predicted Y values were generated for each of the 169 test observations. The fit metric was then calculated using each of these vectors of predictions, and treating the observed Y values in the test set as a 24th set of "predictions".

We will consider multidimensional scaling (MDS) as

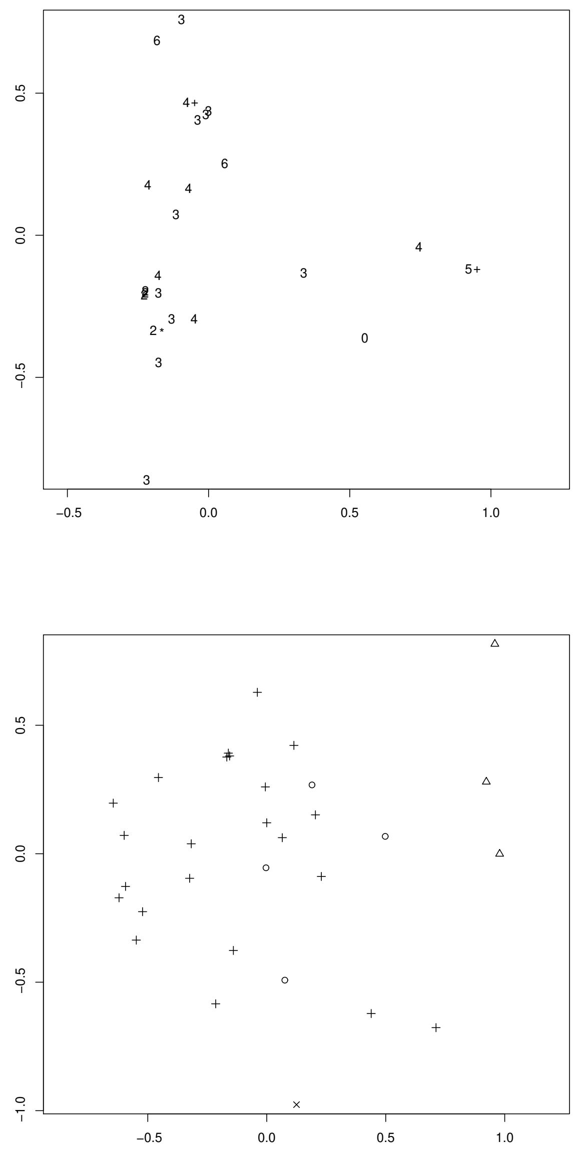

Figure 2: MDS plots for the Boston example. Fit metrics are used. In the top plot, only trees and the test data are displayed, with numbers indicating tree size, and a 0 indicating the test data. In the lower plot, additional models are added, and models are coded as follows: +=treed regression, triangle=conventional tree, o = neural network, x=test data.

a way of representing the dissimilarity matrix. The goal of MDS is to produce a low-dimensional (typically 2) plot such that the inter-point Euclidean distances in the plot approximate the distances specified in the distance matrix. A variety of methods for constructing this mapping are possible. We considered several implemented in R and S-Plus by Venables and Ripley (1999). The `sammon` algorithm (Sammon 1969) produced better mappings than the classical method. Ripley (1996) and many other books on multivariate analysis provide details on these methods. The *stress* criterion was used to identify how good the configuration was. If we have original distances d_{ij} and the MDS configuration gives Euclidean distances $\hat{d}_{ij}$, stress is defined as

$$\text{STRESS} = \sqrt{\frac{\sum_{i,j}(d_{ij} - \tilde{d}_{ij})^2}{\sum_{i,j} \tilde{d}_{ij}^2}}$$

where stress values are interpreted as: 0.20 = poor, 0.10 = fair, 0.05 =good, 0.025=excellent and 0=perfect. For the MDS plot in Figure 2 (top plot), the stress is 0.32, a poor value. Consequently this plot should only be interpreted in very general terms.

In this plot, the number of terminal nodes is used as a plotting symbol, and the observed test data by a 0. We can see that there are several different models that are "close" to the test data, but different from each other. That is, they must be predicting different aspects of the test data well. The two trees displayed earlier in Figure 1 are indicated with a +. They are quite different from each other. The four node tree appears to fit more poorly because of the greater distance from the 5 node tree in the MDS plot. Further examination of the distance matrix revealed the following distances:

	5 node tree	4 node tree
4 node tree	1.173848	0.000000
test values	1.242888	1.173419

In fact these trees are about equally different from each other and the test data, essentially an equilateral triangle. This suggests that if their predictions were averaged, they might predict the test data better. When this is done, the distance between the average fits of these two trees and the test data drops to 1.06, one of the smallest values among all predictions of individual trees.

How were these two trees selected? We want to identify trees that are dissimilar but fit well. In this case we searched the distance matrix for dissimilar trees that had large log likelihoods (evaluated on the training data). Another possibility would be to search the matrix for observations that are close to the test data, but far from each other.

Another tree of interest is the one marked with a *. It corresponds to the two node tree with a city/suburbs split. Its distance from the fitted value is about 1.09 (also distorted in the plot), making it one of the better trees found. The minimum distance from the test set among all 23 trees is 1.05.

The ability of the fit metric to compare models in different classes is demonstrated by adding several models considered in a simulation study (Chipman, George and McCulloch 2000). These include neural networks, and conventional tree models. For the four neural nets, combinations of 3 and 7 nodes with weight decays of 10^{-2} and 10^{-5} were used. Conventional trees of size 5, 10, and 15 terminal nodes were constructed. Details of the logic underlying these parameter choices are given in Chipman et. al. (2000).

Figure 2 (bottom panel) gives the MDS plot with the fit metric for the original trees plus the neural networks, conventional trees, and test data. Conventional trees are separate from other tree models, while treed regressions and neural networks tend to be more similar. This might be due to the piecewise constant nature of conventional tree predictions. Neural nets and treed regressions seem to get quite close to the true response. Again, caution should be exercised in interpreting this plot, as the distances given are nonlinear projections of the actual distances. In this plot, the stress is about .30.

For the purpose of clustering models, it may be useful to consider the quality of the models in addition to the distances between them. Although such quality information could be superimposed on the MDS plot, we propose an alternative that we call the *added model plot*. This plot conveys both the distances between models and their relative quality. The added model plot in Figure 3 uses the distances between trees under the fit metric to assess the effect of adding new trees one at a time from left to right. These trees are added in order of decreasing log likelihood. For each index value on the horizontal axis, all distances between the new tree and all better trees are plotted on the vertical axis. As with MDS, the test data are included as a special "model". Distances to the test data are indicated with a filled circle. Reading from left to right, the second added tree is close to the first (0.45 distance), and their fit is similar. There is thus no compelling reason to examine the second tree since it is similar to the first. A similar argument applies to the third tree. The fourth tree is different from the first three, but doesn't fit as well, while the fifth tree fits well and is not similar to any of the first four. In general we would seek trees that are distant from all others already added and still have sufficiently good fit to be considered for examination.

A natural question to ask about the two trees in Figure 1 is how they differ. Obviously different variables are used, but do the different variables identify similar or different partitions of the data? The partition metric and the data used to calculate this dissimilarity can help answer this. The partition metric for these trees gives a dissimilarity value of .275, meaning that 27.5% of pairs of points are not assigned to the same cluster. This value is large among the 23 trees. A table of the node labels for the 169 test cases is given below:

	A	B	C	D
1	82	13	0	8
2	6	8	12	7
3	1	0	4	12
4	1	2	6	1
5	3	0	1	2

Labels 1-5 for terminal nodes of one tree and A-D for the other tree are used. Nodes 1 and A contain 82 common points (half the test data), while similar but weaker patterns occur for node pairs 2-C and 3-D. Otherwise, the trees are quite different. It would be more reassuring if the trees had similar partitions, but the fact that they don't is good to know before trying to interpret individual trees.

More automated approaches are also possible. Instead of trying to visualize the data (with associated inaccuracies in distances) with a MDS plot, clustering methods might be applied to automatically identify groups of similar tree models. The one additional challenge is to incorporate the "tree quality" (eg log likelihood) into the clustering procedure, as with the added model plot. Chipman et. al. (1998b) do this in a coarse manner by selecting a certain number of "most likely" trees and then clustering them.

5 DISCUSSION

Papers by other authors consider related issues, in the context of tree models. Shannon (1998) looks at predictive accuracy of trees identified as interesting using similar metrics. Hawkins and Musser (1998) use a forest of trees to learn what variables tend to occur together or apart in individual trees. Shannon and Banks (1998) propose the tree metric and use it to construct a single tree that is central to a forest. An important distinction is that they do not rank the trees according to their fit, a central element of our approach.

With the models considered here, the parameters are assumed to be partitioned into two classes: structural and continuous parameters. In other classes of models, the value of this distinction may vary. For example, in neural networks, it is common to use a network with a sufficiently large number of hidden units, and penalize the estimation of weights. There, the structural component (hidden units and connections) is less important than the continuous component (weights). Methods like MARS (Friedman 1991) will be more amenable to the techniques mentioned here, due to the structural parameters which share some similarities with linear

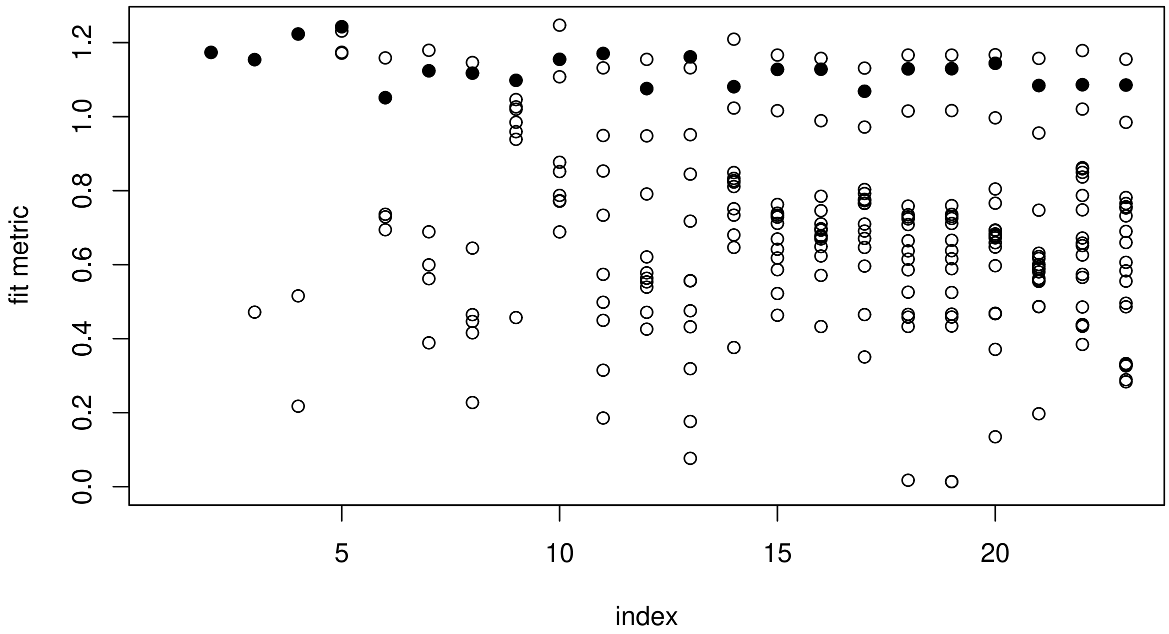

Figure 3: Added model plot for 23 trees, using the fit metric. Trees are ordered from left to right by log likelihood. Distances to the test data are indicated with filled circles.

models and trees.

Other possible uses for the identified clusters exist. In a Bayesian framework, the posterior mass associated with each cluster would give an idea of how likely each group of models are. In many frameworks, model averaging could be simplified by averaging over a few models, one selected as representative of each cluster. The simple averaging of two trees in Section 4 illustrates the promise of this technique, which we will explore in future work.

The emphasis of this paper is post-analysis of output from tree growing algorithms. However, metrics may also be useful in the model construction process. If a very large number of models are to be considered by the model search algorithm, it may be impractical to record all models visited. An interesting alternative would be to use one or more metrics in an "on-line" manner, discarding models that are similar or identical to models that have already been visited. Other possibilities may be interesting; for example the search algorithm could be modified to move in directions that are at least a certain distance from models already discovered.

Acknowledgments

The authors wish to thank the conference organizers and several anonymous referees whose encouraging comments and constructive criticism on an earlier version has lead to improvements in this paper. We also wish to thank David Andrews for suggesting the idea of a partition metric, and Bill Shannon, David Banks, and Bret Musser for interesting conversations. This work was supported by NSF grant DMS 9803756, Texas ARP grant 003658690, the Natural Sciences and Engineering Research Council of Canada, the Mathematics of Information Technology and Complex Systems network, and research funding from the Graduate Schools of Business at the University of Chicago and the University of Texas at Austin.

References

Belsley D.A., Kuh, E. and Welsch, R.E. (1980) *Regression Diagnostics. Identifying Influential Data and Sources of Collinearity*, Wiley, New York.

Breiman, L (1996), "Bagging Predictors", *Machine Learning*, 24, 123–140.

Breiman, L. (1999) "Random Forests - Random Features". Technical report, University of California,

Berkeley.
`www.stat.Berkeley.EDU/users/breiman/`

Chipman, H., George, E., and McCulloch, R. (1998a) "Bayesian CART Model Search (with discussion)", *Journal of the American Statistical Association*, 93, 935–960.

Chipman, H., George, E., and McCulloch, R. (1998b) "Extracting Representative Tree Models from a Forest", working paper 98-07, Department of Statistics and Actuarial Science, University of Waterloo. `www.stats.uwaterloo.ca/~hachipma`

Chipman, H. A., George, E. I, and McCulloch, R. E. (2000) "Bayesian Treed Models", working paper 2000-08, Department of Statistics and Actuarial Science, University of Waterloo. `www.stats.uwaterloo.ca/~hachipma`

Denison, D., Mallick, B. and Smith, A.F.M. (1998) "A Bayesian CART Algorithm", *Biometrika* , 85, 363-377.

Freund, Y. and Schapire, R. E. (1996) "Experiments with a new boosting algorithm", *Proceedings of the Thirteenth International Conference on Machine Learning*, L. Siatta, Editor, 148–156, Morgan Kaufmann, San Francisco, CA.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines", *Annals of Statistics*, 19, 1–141.

Furnival, G. M. and Wilson, R. W. Jr. (1974) "Regression by Leaps and Bounds", *Technometrics*, 16, 499–511.

George, E. I. and McCulloch. R. E. (1993) "Variable Selection Via Gibbs Sampling", *Journal of the American Statistical Association*, 88, 881–889.

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

Hawkins, D.M. and Musser, B.J. (1998) "One Tree or a Forest? Alternative Dendrographics Models", *Proceedings of the 30th Symposium on the Interface.*

Lutsko, J. F. and Kuijpers, B. (1994) "Simulated Annealing in the Construction of Near-Optimal Decision Trees", in *Selecting Models from Data: AI and Statistics IV*, P. Cheeseman and R. W. Oldford, Eds., 453–462.

Ripley, B. D. (1996) *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.

Sammon, J. W. Jr (1969) "A Non-linear Mapping for Data Structure Analysis" *IEEE Transactions on Computers*, 18, 401–409.

Shannon, W. (1998) "Averaging Classification Tree Models", *Proceedings of the 30th Symposium on the Interface.*

Shannon, W., Banks, D. (1998) "Combining Classification Trees using MLE', *Statistics in Medicine*, In Press.

Tibshirani, R., and Knight, K. (1999), "Model Search and Inference by Bootstrap 'Bumping' ", Journal of Computational and Graphical Statistics, 8, 671–686.

Venables, W. N. and Ripley, B. D. (1999) *Modern Applied Statistics with S-PLUS, Third Edition*, Springer.