

Modern Regression Methods: A Comparative Discussion

Kenneth O. Cogger

April 19, 2001

ABSTRACT

In the last ten years, several new piece-wise linear analytical methods have been developed and applied to problems of prediction and function approximation with marked improvement over traditional methods. This progress is important for applications in any field which have in the past used standard linear procedures or their variations. These new procedures should be considered wherever one might be considering discrimination or classification procedures, multiple regression, time series analysis and forecasting, econometric model building, and neural modeling. Examples abound in Marketing, Finance, Accounting, Information Systems, Economics, Engineering, and Mathematics as well as many other sciences. There are differences between these procedures that are not widely understood, and their relationships to more commonly used methods have been discussed only in widely scattered articles in the literature. This paper discusses several of these new developments within a common framework, describes applications and potential applications in various areas, and discusses sources of computer software for interested users. The well-known Boston Housing data is treated with each of three procedures for comparative purposes.

Key Words: Neural Networks, multivariate adaptive regression splines, hinged hyperplanes, classification and regression trees, adaptive logic networks, recursive partitioning.

The author is President, Peak Consulting, Inc. and Professor Emeritus, University of Kansas.
32154 Christopher Lane, Conifer, CO 80433
cogger@peakconsulting.com

1. Introduction

Recently, several new piece-wise linear analytical methods have been developed. With computer software becoming more widely available, these nonlinear methods have been applied with varying degrees of success. In this paper, the newer techniques discussed include multivariate adaptive regression splines (MARS) (Friedman[1991]), hinged hyperplanes (HHP) (Breiman[1993]), and adaptive logic networks(ALN) (Armstrong[1995]). To understand their relative strengths and limitations, we also mention some older nonlinear techniques which are perhaps more familiar, including classification and regression trees(CART) (Breiman et al. [1984]) and artificial neural networks(ANN).

All of these techniques may be applied to problems of prediction and function approximation, where one wishes to model the dependence of a response(dependent) variable y on one or more predictor(independent) variables which for convenience we collect in a column vector \mathbf{x} . Standard regression problems fit this description and in fact are a special case of each of the piecewise linear methods discussed here. As such, the general linear model is a suitable null case for measurement of improvement. Beyond typical regression applications, these procedures may also be applied to binary classification problems as well as time series fitting and prediction. In the latter situation, a suitable null comparison may be ARIMA models (Box, Jenkins, and Reinsel[1994]).

The organization of this paper is as follows. In Section Two, descriptions of three competitive techniques are given. Known sources of computer software are also given for those readers interested in trying some of these methods on their own data. These software sources are probably incomplete, as some were found in obscure locations and the rapidly developing nature of this field will render any such list incomplete very quickly. The author would appreciate knowing of any other sources not listed. The discussions are brief, but sufficient for the reader to discern what each technique produces in the way of a model, how this is done, and some of the underlying details that are necessary for understanding later discussions of differences in the capabilities of each method. Section Three summarizes the methodological distinctions that can affect performance of each method and in some cases quite severely limit modeling power. This discussion should serve to guide practitioners in choosing which method to apply in any given situation. Section Four discusses actual and potential applications in a number of areas. This discussion is meant to be thought-provoking and to stimulate the reader to begin to think creatively about the usefulness of these nonlinear techniques in their own area of expertise. Section Five reports on the application of each procedure to the Boston Housing data, which has been the subject of much analysis in the regression, machine learning, and decision tree literature. Our general conclusions and recommendations appear in Section Six.

2. The Methodological Basics

2.1 Adaptive Logic networks(ALN). An ALN model is a tree with any number of layers, where the leaf layer contains linear combinations of the form $\mathbf{b}'\mathbf{x}$, where \mathbf{b} is a column vector of parameters (which may be viewed as regression coefficients) and \mathbf{x} is the vector containing values of the input or predictor variables. Usually, the first component of \mathbf{x} equals 1 to allow for a constant term or 'bias' as it is known in neural network applications. Each such hyperplane will be referred to as a linear 'piece'. For the next layer in the tree, node values are calculated as either the Minimum or Maximum value of one or more leaf nodes. The tree may be binary, with only two inputs to each node, but this is not necessary. Subsequent layers are calculated in a similar fashion until the final layer, which has a single output node to compute the ALN model's scalar output y . An example of an ALN to describe a piecewise linear non-monotonic function is given in Figure 1.

It is easily demonstrated that an ALN with suitable architecture(layers, nodes, and node types) and coefficients (the weight vectors in the leaf layer) can approximate any continuous function to any desired accuracy with a piece-wise linear approximant. The word "logic" in ALN can be motivated from a couple of perspectives. First, Max and Min correspond directly to the logic functions And and Or. For example, $(y < \mathbf{a}'\mathbf{x} \text{ And } y < \mathbf{b}'\mathbf{x})$ if and only if $y < \text{Min}(\mathbf{a}'\mathbf{x}, \mathbf{b}'\mathbf{x})$. Similarly, Or and Max correspond directly. Second, "logic" is seen in the flow of control during the numeric evaluation of an ALN, since entire subtrees of an ALN can be omitted from formal evaluation based on comparing the input values at nodes. The simple Max-Min structure thus makes an ALN computationally very efficient compared to other nonlinear

methods. (Software for ALN actually uses linear threshold functions for computational purposes, with Boolean valuation of leaf nodes such as $(y - \mathbf{a}'\mathbf{x})?$). This is similar to alpha-beta pruning in games. When creating the And or Or of such logic functions, many branches need not be formally evaluated; if a node is And and any input is False, none of the other inputs need be evaluated. To evaluate the numeric output y , one notes that the responsible linear piece lies on the border between 'True' and 'False' space. Unlike many other techniques, including neural net models, typically only a small number of input functions need be explicitly evaluated for any given input vector \mathbf{x}).

The power of an ALN can only be utilized properly if an appropriate architecture (number of layers, nodes in each layer, and node types) and coefficients (weight vectors) are determined. Current software utilizes an adaptive iterative least squares technique similar in many ways to the backpropagation technique employed in artificial neural networks. A RMSE output error tolerance T is prescribed. Starting with a single piece, which is equivalent to standard multiple regression, the coefficients in the (single) weight vector are estimated to minimize RMSE. If $\text{RMSE} > T$, the piece is broken into two pieces which adapts to either a Max or Min function, whichever is most appropriate. Iterative least squares allows the break point to be estimated along with all coefficients. Each piece, then, if responsible for a prediction error, adapts by altering its coefficient vector \mathbf{b} . Splitting stops when all pieces have $\text{RMSE} < T$.

Clearly there is room for overfitting of the model if T is too small, so it is advisable to utilize protective techniques such as having a training and validation sample to ensure that T is chosen appropriately and then testing the final model on a second holdout sample.

One important feature, common to some of the other piece-wise linear approaches discussed later, is that because the underlying pieces of the model are linear, analyses of variance (ANOVA) and standard t -tests and F -tests are, at least in large samples, familiar and reasonable measures for determining significance of variables, the importance of individual pieces, etc. The author has also developed statistical procedures for holdout samples which do not require large sample sizes. A second desirable characteristic is that because for any input vector \mathbf{x} , only one linear piece will be responsible for computing the output y , the interpretation of the model is very easy. Coefficients represent partial derivatives, simultaneous (small) change of two or more inputs can be examined, elasticities and other familiar terms used in linear regression can be employed with ALN's. It should be noted that the fitting of some functions, particularly if they involve inflection points, may require 'artificial' pieces with large coefficient values for which few if any input vectors \mathbf{x} are assigned; such pieces serve only to separate convex from concave regions of the surface.

The author has actually set up ALN coefficient estimation for some simple ALN structures inside Microsoft Excel spreadsheets, so it is possible to use ALN's without proprietary software. However, currently, software for the general ALN model is available free from Dendronic Decisions Ltd at www.dendronic.com. Programs in C++ may utilize their Dendronic Learning Engine to customize applications and output.

2.2 Hinged Hyperplanes. An HHP model is also a piecewise linear model, and has also been proven to approximate any smooth function arbitrarily closely with a sufficiently large number of pieces. A hinge function consists of two hyperplanes and any hinge function is either the Max or the Min of $(\mathbf{a}'\mathbf{x}, \mathbf{b}'\mathbf{x})$ where all vectors have the same interpretation as in the discussion of ALN's. The difference between an ALN and an HHP model is that the former utilizes a tree structure, while the general HHP output is the simple sum of a number of hinge functions, each with either a Max or Min operator, and each with two unique hyperplanes. (If the number of hyperplanes in the model is odd-numbered, this can be accommodated by having a hinge function with a zero element such as in $\text{Max}(0, \mathbf{a}'\mathbf{x})$.) A simple HHP is illustrated in Figure 2 with this property. The underlying function is the same in Figures 1 and 2, to allow a direct comparison of the similarities and slight difference between an ALN and HHP representation of the same mathematical function. The reader should note that the hyperplanes explicitly represented in the ALN and the HHP are not identical. Most of the hyperplanes in an HHP represent differences between potentially active linear pieces; all of the hyperplanes in an ALN are the potentially active hyperplanes themselves.

Computational efficiency of an HHP may be less than an ALN, even for the same functional output, since every hyperplane in every hinge function must be evaluated to compute the output y . Some function approximations in ALN's may require, as noted above, extra 'artificial' pieces, so this feature of ALN's may make the comparison a bit closer and these dummy pieces may initially cause some difficulty in interpretation until it is recognized they don't involve many, if any, input vectors \mathbf{x} .

HHP estimation starts with an arbitrary hinge location, estimates the two hyperplanes with ordinary least squares and then computes a new hinge by taking the difference between the two estimated

hyperplanes; this is repeated until these differences, which are estimates of the hinge location, no longer change by a significant amount; this tolerance is similar to that in ALN's and must be specified by the user. Simulation studies support the conclusion this convergence is very fast even with high dimensional input vectors in the presence of high noise levels. To avoid overfitting, the usual validation techniques can be employed or certain penalty functions employed on the squared error to reflect the increased number of parameters as the number of hinge functions increases. Additional hinge functions can be added incrementally by using the basic algorithm on the prediction errors and then readjusting all the hinges.

Since linear pieces are the building blocks of an HHP model, it is possible to construct the usual ANOVA measures for each piece, but interpretability is somewhat hampered by the fact that any input vector affects all the hinge functions; locally, in an ALN, only a few pieces are relevant and usually only one piece. Out-of-sample statistical inference, as mentioned above for ALN's, is probably feasible for HHP models as well, but the author is unaware of developments in this direction.

The author is also not aware of commercially available software that implements HHP, although it would probably not be difficult to put together a rudimentary program with all the basic features perhaps even in an Excel worksheet using a Visual Basic macro to carry out the procedures for the general HHP algorithm described by Breiman. Simple HHP models can certainly be implemented in such a manner, and that is the approach employed on the Boston Housing data reported in Section Five.

2.3 Multivariate Adaptive Regression Splines. MARS was first described ten years ago and has received a great deal of attention since, although perhaps not as much as deserved due to the lack of user-friendly software packages. This situation has now resolved with the introduction of the first commercial MARS software (www.salford-systems.com). XTAL incorporates a version of MARS for SUN workstations (www.ece.umn.edu/groups/ece8591/xtal.html).

A MARS model fits separate basis functions, or splines, to distinct intervals of each of the predictor variables in \mathbf{x} . In MARS, the basis functions are of the form $\text{BF}(x) = \text{Max}(0, x-k)$ and $\text{Max}(0, k-x)$ where x is any of the predictor variables and k is a parameter which must be estimated. The simplest MARS model is a linear combination of the form $y = \mathbf{a}'\mathbf{BF}(\mathbf{x})$, where \mathbf{a} is a coefficient vector and $\mathbf{BF}(\mathbf{x})$ is a column vector containing terms of the indicated form. Since $\text{Min}(0, x-k) = -\text{Max}(0, k-x)$, with suitable algebraic signs on the coefficients in \mathbf{a} , the output y of the simplest MARS model is always mathematically equivalent to the simple sum of a series of Max and Min functions. This is a special case of an HHP, since each term in the model is a simple linear function of a single input x rather than a hyperplane involving the entire input vector \mathbf{x} . It is important to note that more complicated MARS models incorporate interaction terms which employ the product of the simple linear threshold basis functions. An open question is whether the inclusion of simple linear threshold functions and their products can handle multicollinearity in a regression context, since linear combinations of the input variables cannot be incorporated in a MARS model except by choosing the 'knot' or 'hinge' value for each input x at the minimum or maximum value of its range leading to the special case $\mathbf{BF}(\mathbf{x})=\mathbf{x}$. An example of a simple MARS model in the univariate case is given in Figure 3. Again, note the difference in how the elements of the model are interpreted.

As with HHP, the computation of y for a MARS model requires the explicit evaluation of every basis function. This may be mitigated by the allowance of interaction terms, but the aforementioned difficulty in allowing for explicit linear combinations of all variables may be an important limitation.

In the univariate case, MARS, HHP, and ALN models are all capable of producing exactly the same output functions. All are piecewise linear and continuous. In higher dimensions, however, the situation becomes more complicated. HHP and ALN models do not explicitly allow for interaction terms. Interaction terms in HHP and ALN are not precluded, but they must be introduced mechanically by including products of input variables as part of the input vector \mathbf{x} . In MARS, at least in the commercial software version, these interactions can automatically be included with the click of a button. This is a convenience that may or may not be important depending on the application.

The difference between MARS and HHP and ALN in the basic situation without interaction terms is extremely important to understand. Examining the basis functions in MARS, it is clear that changes in the form of the output function occur only at the hinge points (or knots) $x=k$, which involve only one variable at a time. This means that the hinges in MARS are restricted to be parallel to the coordinate axes of the input variables. HHP and ALN models are not so restricted. In this sense, MARS retains the limitations of CART, which in most implementations restricts its recursive partitioning to be univariate. In the univariate input vector case, this restriction is meaningless, but in higher dimensions, it means that a function such as $y=\text{Min}(\mathbf{a}'\mathbf{x}, \mathbf{b}'\mathbf{x})$ can be fit perfectly by either an HHP or ALN with two linear pieces, but MARS will be forced into approximating this function with, theoretically, an unboundedly large number of linear pieces

formed by adding a large number of basis functions (unless the coefficient vectors \mathbf{a} and \mathbf{b} happen to satisfy very tight restrictions).

While there are weaknesses in the flexibility of the MARS model, these may in some applications be overcome by the strength of its estimation procedure. MARS potentially tests every value of every input variable as a potential value of a knot/hinge point. This will lead to a most likely vastly overfit model. This model is then pruned back in the MARS algorithm using generalized cross-validation techniques which do not require use of separate test, validation, and holdout samples, eliminating basis functions one at a time until the RMSE, penalized for model complexity, is minimized. Particularly in large data mining applications, this automated feature of MARS may well offset the deficiencies in the model itself. If large sample sizes are extant, of course, it is certainly wise to test the final MARS model on a true holdout sample to verify that it has generalized well on out-of-sample data.

2.4 Other Nonlinear Techniques. Classification and Regression Trees(CART) have been available for a longer period of time than ALN, HHP, and MARS. They have been used in a wide variety of applications and are by now fairly well understood. Breiman, et al.[1984] is the basic reference for CART, and many researchers now believe that with the advent of MARS that for continuous valued outputs y , there is little reason to employ CART except possibly as a preliminary exploratory device, examining whether, for example, certain interaction terms suggest themselves or whether it might be prudent to eliminate the added complexity of interactive terms in the MARS approach.

Clearly CART uses binary partitioning, resulting in the same weakness as MARS, but also the end model uses a predicted y that is simply the average y in each rectangular subspace. Such a predictive function is not even continuous, making CART a poor choice in some applications, such as control engineering where the output function must be continuous. MARS, HHP, and ALN all produce continuous predictive functions, which are necessary in many applications, and likely to give lower RMSE as well.

Generalized Additive Models(GAM) are the most general form of a likelihood-based regression model. GAM is based on replacing the coefficients found in standard regression models by smoothing functions, which may be localized averages, localized linear regressions, splines, etc. The response y is then assumed to come from a probability distribution that is in the general exponential family (the normal distribution e.g.). The results one obtains with GAM are apt to depend strongly on the choices made for the type of smoothing function, the degree of smoothing, and the judgment of the user. If localized regression is used as the smoothing technique, and a normal distribution is assumed, GAM will produce models of the same form as HHP, ALN, and MARS, but the added complexity of the many choices that must be made by the user would seem to favor the latter techniques. Miller[1999] compares GAM, MARS, and CART on the well-studied Boston Housing Data (Harrison and Rubinfeld[1978]) and observed that GAM had poor predictive performance with small sample sizes; MARS and CART performed well across a wide range of sample sizes.

Traditional artificial neural networks(ANN) (Cogger[1997] and many other summaries on the internet, e.g.) have been used in a variety of applications for the last twenty years or so. They have been particularly successful in applications where highly nonlinear input-output relationships are suspected. However, piecewise linear techniques such as ALN, HHP, and MARS are proving to be at least as good at function approximation and prediction as traditional ANN's in many applications. In fact, the usual definition of ANN allows these techniques to be thought of as special kinds of ANN's. A major practical consideration in many applications is the difficulty in interpreting any final predictive ANN model, and potential problems in function fitting. In economic and business applications, prediction may be important, but understanding the fitted model may be even more important. For example, in a large data mining application investigating consumer buying behavior; it may be just as important to understand why a consumer is predicted to be a purchaser as it is to simply predict that they may be a likely purchaser. Piecewise linear models offer this ease of interpretation. Each piece of the model can be explained in terms that are familiar in a simple regression context. Rates of change, elasticities, etc. are describable even to people not familiar with statistics. Traditional ANN's produce more of a "black box" kind of input-output association. There is another issue with traditional ANN's that may not be obvious to the casual observer. In chemical process control, among other application areas, safety may be an important consideration. One wants to be assured that the fitted function in such situations has no sharp discontinuities or near discontinuities. With the piecewise linear functions employed by ALN, HHP, and MARS, such conditions are impossible. With ANN's, there is no assurance even after thorough testing that there is not some anomalous condition. This is particularly true in high dimensional data spaces.

3. Methodological Distinctions

The three main nonlinear methods discussed here all produce piece-wise linear models. This means that the general linear model widely used in practice is simply a special case of each method. The methods are distinguished by their modeling flexibility, ease of use, and software availability. ALN and HHP are the most flexible, with proven ability to model any smooth response function, albeit with a large number of pieces depending on the nature of the function. (The function $y = x^2$ for example) If the function is truly piece-wise linear, MARS will probably require a larger number of pieces than either ALN or HHP. It has been shown that for some simple functions, ALN and HHP will fit the function exactly with two pieces and MARS will require very many pieces to approximate the function.

MARS is probably easier to use than HHP or ALN, primarily because the modeling steps have been fully automated, although some parameter choices must be made by the user, and decisions must be made regarding whether to include interaction terms. The inclusion of interaction terms is possible in ALN and HHP, but is not automated.

The tree structure of ALN makes it computationally efficient in applications where speed is important. A possibility that has not been discussed very often is to estimate a piecewise model with whatever software one desires: MARS, HHP, etc. and then convert the final model to the ALN tree structure and take full advantage of its computational speed. In applications where speed is essential, such as nuclear power station control, identification of deorbiting space objects, etc., this translation may be of high value.

Translation of MARS and HHP models to ALN equivalents has not been described in the literature as far as the author knows. But it turns out that this translation is feasible using some basic properties of $\text{Min}()$ and $\text{Max}()$ functions. First, a final MARS model is a special case of HHP if one incorporates cross product terms from the former into the latter. The essential result, then, is to show that any HHP model may be translated into an ALN. First, note that the following identities hold:

$$\text{Max}(a, b) + \text{Max}(c, d) \equiv \text{Max}(\text{Max}(a + c, a + d), \text{Max}(b + c, b + d))$$

$$\text{Max}(a, b) + \text{Min}(c, d) \equiv \text{Max}(\text{Min}(a + c, a + d), \text{Min}(b + c, b + d))$$

$$\text{Min}(a, b) + \text{Min}(c, d) \equiv \text{Min}(\text{Min}(a + c, b + c), \text{Min}(a + d, b + d))$$

(Note: The middle identity has an alternate form in which the right hand side is the Min of two Max's)

Second, it is clear that an HHP with a single hinge is identical to an ALN with two linear pieces. Adding either a $\text{Max}()$ or a $\text{Min}()$ to a simple HHP produces an ALN with four pieces as a tree nesting a series of Max and Min operations. Adding another $\text{Max}()$ or $\text{Min}()$ function to the left hand side of any of the above identities, by extension, will produce another ALN with deeper binary nesting of Max and Min operators. Any HHP, therefore, can be represented ultimately as an ALN. Note that these equivalent ALN trees may have few layers and thus be simpler than they appear at first glance. For example, the right hand side of the first identity above may be expressed as

$$\text{Max}(a + c, a + d, b + c, b + d)$$

which is a tree with a single layer and four leaf nodes.

Another distinction between the methods that may be important is the availability of software. Free software is available for ALN (<http://www.dendronic.com>). MARS software is available in commercial form (<http://www.salford-systems.com>) as well as in the public domain (for S-PLUS: <http://lib/lib.stat.cmu.edu>). MARS for SUN work stations has been cited previously. The author is not aware of public domain software for HHP.

4. Applications: Actual and Potential

In **Marketing**, it is of great importance to be able to identify customers and potential customers who are more (or less) likely to purchase a product and also to predict frequency of purchase based on customer

characteristics, past purchasing behavior, etc. that information often being available in huge databases created from, e.g., bar scanning at retail outlets. It is arguably unlikely that a linear model would be able to tease complicated buying behavior relationships out of such large databases. Nonlinear approaches as described above are all capable of automating very sophisticated search strategies for building predictive models in these situations.

In **Finance**, models such as generalized autoregressive conditional heteroscedasticity (GARCH) (Bollerslev[1986]) have become standards for analyzing financial market and other economic fluctuations. Outputs y from these models are inherently nonlinear in their inputs x , yet rest on strict assumptions about the generating process. If these assumptions are violated, there is a reasonable argument that more general nonlinear mechanisms as discussed in this paper might well have higher predictive ability. Several mutual funds (e.g. Disciplined Equity of the Vanguard family of funds) are known to base part of their portfolio selection strategy on neural nets. Yet some of the piecewise linear techniques discussed here are worthy competitors of neural nets. Forecasting foreign exchange rates even in volatile markets has been successfully accomplished with ALN's, for example (Cogger et al.[1997]).

In **Accounting**, the use of ALN's to predict the occurrence of fraudulent management behavior in auditing has been shown to be superior to standard classification techniques and neural nets (Fanning and Cogger[1995]). It is widely believed that the selection of federal tax returns for human examination is a procedure that partially uses neural nets; however their complexity and the clarity of interpretation possessed by some of the newer nonlinear techniques discussed here may argue for examining them in this context, which is quite similar to the audit fraud context.

In management information systems (**MIS**) a great deal of information is collected, often without useful means for interpreting, summarizing, and explaining that information. In this area as in marketing, data mining techniques have often been suggested as potentially useful, arguing again for an examination of some of the newer nonlinear techniques.

In **Economics**, many models are constructed with essentially linear tools. Large econometric models of the U.S. and individual states' economies are employed by the Congressional Budget Office, the Board of Governors of the Federal Reserve System, and the executive branch to assist in policy making. The recent availability of user-friendly software for handling large data sets with nonlinear methods and their increased ease of interpretation over the "black boxes" of neural net models should allow improvements in these econometric models. A recent paper (Cooper[1998]) has employed CART models to examine whether U.S. output fluctuations behave nonlinearly. Given the limitations of CART noted above, this seems to suggest the time is ripe to revisit this data with some of the more modern techniques outlined in this paper.

In **Medicine**, most disease detection and prognostication is accomplished with the simplest nonlinear model: the threshold or step function. Examples include diastolic blood pressure above some threshold, cholesterol exceeding some level, etc. Yet it may be highly nonlinear combinations of individual test results that really excel at disease detection or in assessing the prognosis of an individual with known serious disease.

In **Engineering**, ALN's have been used successfully to predict pavement durability based on various site, material, and application variables. Attoh-Okine[1999] has developed the use of these and other nonlinear modeling approaches in predicting, basically, potholes.

In the search for extraterrestrial intelligence (**SETI**) project, fast Fourier transforms and other techniques are employed in the analysis of radio telescope data. But Fourier transforms are based on autocorrelations, which themselves measure only linear associativity, again suggesting the potential for thinking about nonlinear procedures.

In time series analysis, which is used widely in government, industry, and business, ARIMA modeling is a standard analysis used for prediction and process control. Threshold autoregressive (TAR) models were developed some time ago (Tsay[1989]) and shown to be superior for modeling some time series. TAR models are, however, mathematically equivalent to the simplest two-component HHP, MARS, and ALN structures. At least one recent paper has explored expanding such time series applications in the case of MARS, calling it TSMARS. See Lewis and Ray[1997] for an application to fluctuations in sea surface temperatures. Models identical to those of TSMARS can be generated with ALN and HHP by simply letting the input vector x contain lagged values of the time series output y .

These are just a few of the potential applications where it seems clear there is plenty of room for potential improvement with some of the newer nonlinear models.

5. Analysis of the Boston Housing Data with ALN, HHP, and MARS

The Boston Housing Study(Harrison and Rubinfeld[1978] examined the influence of 13 variables on the median value of homes in 506 census tracts in that metropolitan area. The purpose of the study was to examine the influence of pollution (measured by nitrogen oxide concentration) on housing prices, adjusting for the effects of other variables such as distance from work centers, tax rates, industrial concentration, location, demographic variables, etc. Their final predictive model employed various nonlinear transformations (logs, squaring) of both input and output variables. The raw data has been in the public domain for some time, and has therefore been the subject of further analysis with CART, consideration of potential outliers and overly influential cases, and even its examination in machine learning exercises.

We revisit the Boston Housing Data by fitting predictive models based on ALN, HHP, and MARS. For a single data set such as this, no conclusions should be drawn about the relative predictive ability of these procedures. Indeed, the inherent differences in the training algorithms and the many choices required of the user in terms of cross validation measures, the use of holdout samples, etc. precludes direct comparisons. But it is interesting that all three techniques exhibited marked improvement over the standard linear model applied to this data.

Our study design was as follows. First, we split the sample of 506 census tracts randomly into a training/validation sample of size 298 and a holdout sample of 208. Training and fine-tuning of each model utilized the training/validation sample which, when required, set aside approximately 30%, or 90 cases, for internal validation. This splitting results in the training and holdout samples being roughly equal in size at 208 cases each. Miller(1999), in an analysis of this data set, concluded that such a split of 'learning' and 'test' samples offered reasonable protection against over-fitting and lack of generalization except in the case of Generalized Additive Models, which did not generalize well with (learning) sample sizes greater than 300.

The ALN model was selected by setting a preliminary tolerance, fitting the model with iterative least squares, minimizing RMSE, and then adjusting this tolerance until the RMSE in the validation sample gave indications of good generalization from the training sample to the validation sample. An available option using this software was the bagging technique, where seven models are fitted to improve generalization.

The HHP model was estimated by minimizing RMSE in the training sample, starting with a single regression plane, and then adding pieces incrementally until RMSE in the validation sample exhibited no further improvement. The algorithm employed was gradient search with binary constraints as needed. This produced a final model with three linear pieces, all hinges being of the Max() variety.

The MARS model used the commercial version 2.0 with no penalty chosen for added variables, a maximum of 15 basis functions, and the entire training/validation sample of 298 census tracts. The results reported below are based on no interactive terms. We did explore the use of interactions on this data, but did not find improved performance in the holdout sample, although results were much improved in the training/validation sample. In determining the final MARS model, we used the available software option of setting every third case aside randomly for a validation sample. In this sense, the training/validation sample was utilized in approximately the same manner as for the ALN training.

The table below gives summary results for each method applied to the Boston Housing data. By comparison, we have included the results for a standard multiple regression analysis of the same data. The number of linear pieces for each technique is also reported. As proven above, any MARS or HHP model has an ALN equivalent. The differences in the estimated models reported here are due to the different training algorithms employed, differing parameter settings by the author, etc. And, since these analyses employed a single data set with a fixed, non-replicated assignment into training, validation, and holdout samples, the reader is further cautioned against drawing performance comparisons among the techniques.

Model:	Regression	ALN	HHP	MARS
RMSE(T)		2.002	2.453	n.a.
RMSE(V)	4.164 (T+V)	3.085	3.777	2.911
RMSE(H)	6.213	4.856	10.580	12.801
#pieces	1	~12	3	13

6. Conclusions

This paper has described some of the more recent developments in piecewise linear modeling that show great promise in improving our ability to develop predictive equations and functional approximation. All of them exhibit sufficient flexibility to compete with less interpretable models such as neural nets, yet be decomposable into linear components that may be statistically tested and interpreted in familiar ways. Many have software available that is suitable for large data sets, permitting their use in a wide range of practical problems facing practitioners in many fields. We encourage their consideration given the continuing reports of their successes in practice. Since some of the methods have theoretical limitations, we have noted these in our discussions and urge practitioners to use caution when applying them.

REFERENCES

- Armstrong, W.W. and Thomas, M.M.(1995), "Adaptive Logic Networks", **Handbook of Neural Computation**, Oxford University Press, Section C1.8
- Attoh-Okine, B.(1999) "Flexible Pavement Roughness Prediction Using Adaptive Logic Networks", **Journal of Smart Engineering Design**, October.
- Bollerslev, T.(1986) "Generalized Autoregressive Conditional Heteroskedasticity", **Journal of Econometrics**, vol. 31, 307-327.
- Box, G.E.P., Jenkins, G.M., and Reinsel, G.C.(1994) **Time Series Analysis Forecasting and Control**, 3rd edition, Prentice Hall.
- Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J.(1984) **Classification and Regression Trees**, Wadsworth: Belmont, California.
- Breiman, L.(1993) "Hinging Hyperplanes for Regression, Classification, and Function Approximation", **IEEE Transactions on Information Theory**, vol. 39, no. 3, 999-1013.
- Cogger, K.O.(1997) "A Primer on Neural Networks", **AI/ES Update**, American Accounting Association, vol. 6, no. 2, 3-6.
- Cogger, K.O., Koch, P.D., and Lander, D.M.(1997) "A Neural Network Approach to Forecasting Volatile International Equity Markets", **Advances in Financial Economics**, vol. 3, 117-157.
- Cooper, S.J.(1998) "Multiple Regimes in U.S. Output Fluctuations", **Journal of the American Statistical Association**, vol. 16, 92-100.
- Fanning, K., and Cogger, K.O.(1995) "Detection of Management Fraud: A Neural Network Approach", **Intelligent Systems in Accounting, Finance and Management**, vol. 4, 113-126.
- Friedman, J.H.(1991) "Multivariate Adaptive Regression Splines", **The Annals of Statistics**, vol. 19, no. 1, 1-141.
- Harrison, D., and Rubinfeld, D.L.(1978) "Hedonic Housing Prices and the Demand for Clean Air", **Journal of Environmental Economics and Management**, 5, 81-102.
- Hastie, T.J., and Tibshirani, R.J.(1987) "Generalized Additive Models", **Statistical Science**, vol. 1, 297-318.
- Lewis, P.A.W., and Ray, B.K.(1997) "Modeling Long-Range Dependence, Nonlinearity, and Periodic Phenomena in Sea Surface Temperatures Using TSMARS", **Journal of the American Statistical Association**, vol. 92, no. 439, 881-.893.
- Miller, T.W.(1999) "The Boston Splits: Sample Size Requirements for Modern Regression", **Proceedings**, Statistical Computing Section of the American Statistical Association, 210-215.
- Tsay, R.S.(1989) "Testing and Modeling Threshold Autoregressive Processes", **Journal of the American Statistical Association**, vol. 84, 231-240.

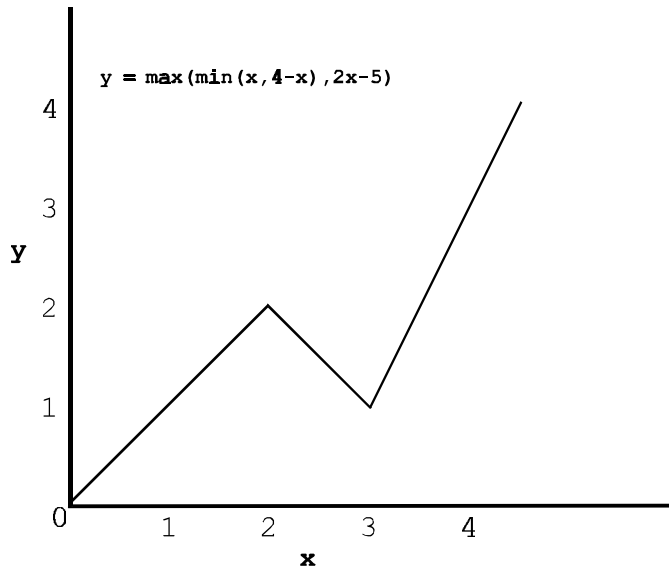


Figure 1 Adaptive Logic Network Example

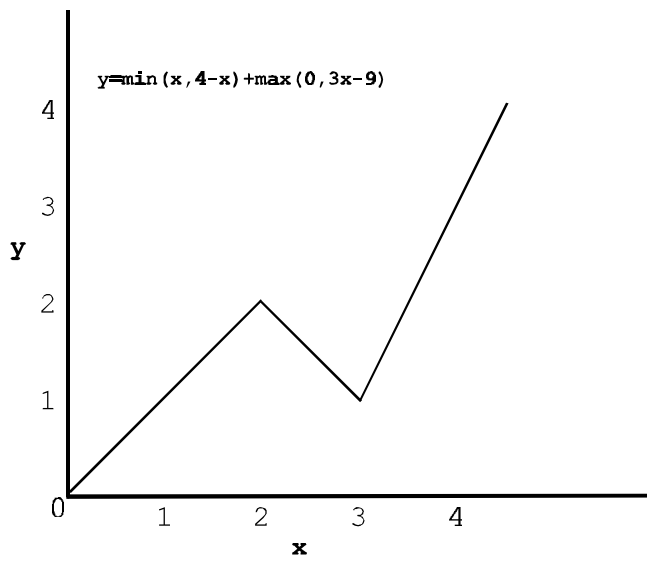


Figure 2 Hinged Hyperplane Example

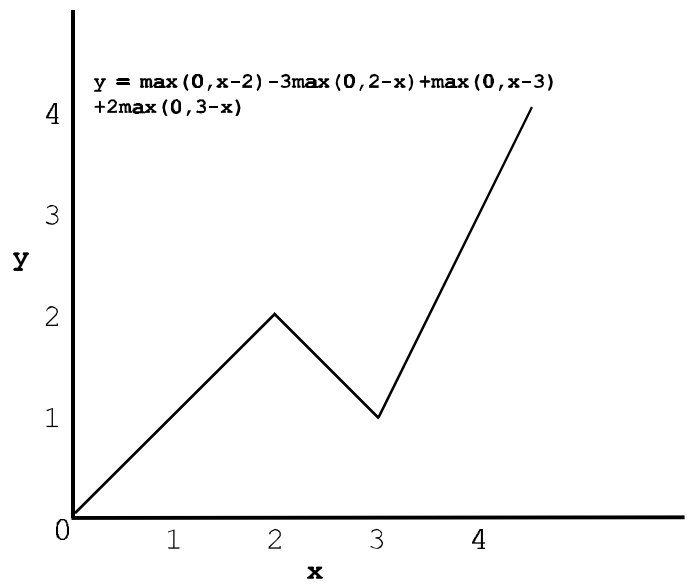


Figure 3 MARS Example