A DISTANCE METRIC FOR CLASSIFICATION TREES

William D. Shannon
Washington University School of Medicine
Division of General Medical Sciences
660 S. Euclid Ave., Campus Box 8005
St. Louis, MO. 63110
shannon@osler.wustl.edu

David Banks
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

CART is an unstable classifier resulting in significant changes in tree structure for small changes in the learning set (Breiman, Friedman et al. 1984; Breiman 1994). To address this problem, research into combining classifiers has increased significantly in the last few years (Breiman 1992; Wolpert 1992; Breiman 1994). These methods are of two basic types: concatenation uses the output from one classifier as input to the next classifier; parallel classifiers work on the same input data with the output from each classifier combined using regression or vote–counting techniques (Schurman 1996). These strategies greatly improve the predictive power of unstable classifiers.

However, when the goal of the statistical analysis is to learn about the relationship between outcome and predictors, these strategies for combining classifiers are unacceptable since they produce a large number of trees, making interpretation difficult. We present a new method for combining classification trees which results in a single, interpretable tree. We begin by defining a distance metric between two trees based on the amount of rearrangement needed so that the structure of the two trees is identical. Using this distance metric, we develop the concept of the central, or median, tree structure and estimate it using a consensus rule. This tree is seen to be more centrally located than the tree fit to all the data. We finish by discussing future work including alternative methods for estimating the median tree, probability models, uses in data mining and meta–analysis, and performance measurements of the median tree.

MOTIVATION

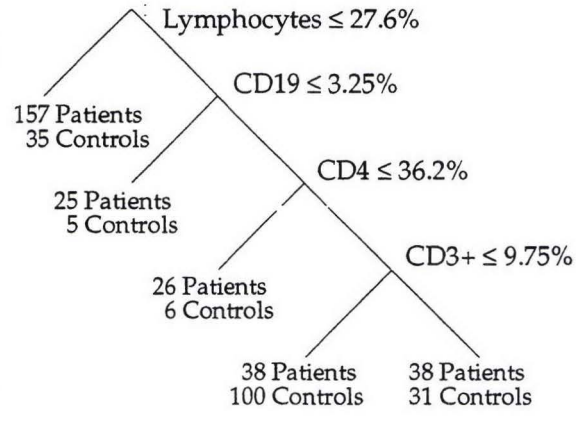
Data from 13 cancer clinical trials were combined into a single data set. The variables measured in each trial were related to immunological functioning, and all patients (N=283) had advanced stages of various forms of cancer. In addition to patient data, immunological data had been collected

for laboratory quality control purposes on 177 normal, healthy control individuals. Figure 1 shows the classification tree generated to predict health status of individuals (cancer patient versus normal control) based on the immunological data using CART software (California Statistical Software, Inc., Lafayette, California) with the Gini criterion for splitting and 10–fold cross-validation for pruning. The classification tree starts with the root node at the top containing the entire data set of 283 cancer patients and 177 normal controls. The first partition splits the data on lymphocyte count: cases with a lymphocyte count less than 27.6% (cut point) fall into the left subset (child node). Of these people, 157 (82%) were cancer patients, and 35 (18%) were normal controls. Since this node is not further partitioned these cases are predicted to be cancer patients. Observations in the root node with lymphocyte count greater than or equal to 27.6% fall into the right subset (child node). The right child node is further partitioned into two subsets (its left and right child nodes) based on CD19 count being above or below 3.25%. Subsets may then be recursively partitioned in a similar manner until further partitioning does not produce a significant improvement in fit. Formal methods for performing recursive partitioning, selecting variables and cut points at nodes, and deciding when to stop partitioning are well described in the statistics (Breiman, Friedman et al. 1984; Clark and Pregibon 1992) and machine learning (Quinlan 1993; Langley 1996) literature, and will not be discussed in this paper.

The classification tree in Figure 1 split on four immunological parameters in biologically meaningful ways. Low counts of lymphocytes, natural killer cells (CD19) or T–cells (CD4) is indicative of a weakened immune system, an expected state in someone with advanced stages of cancer. When these variables are high, an increased activated T–cell (CD3+) count indicates the body is fighting a disease, such as advanced cancer. This tree has an overall cross–validated misclassification rate of 28%.

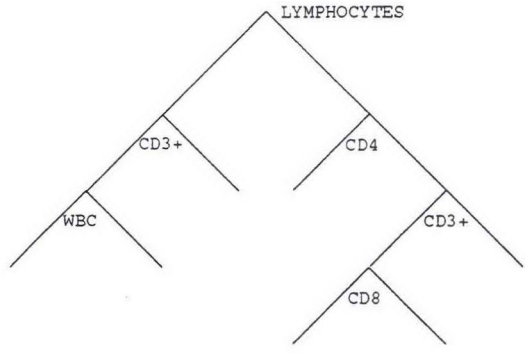
We were concerned that the tree structure in Figure 1 might be the result of data from one, or a few, specific trials. To get a sense of how each trial influenced the final tree, we used a jackknife approach where by we removed the data for a single trial from the learning set, and fit a classification tree to the remaining data. Using this 'leave–one–dataset–out' method, we generated 13 classification trees, each one representing the removal of a different trial from the data set. These 13 classification trees had misclassification rates similar to the tree fit with all the data. On visual inspection, however, the

Figure 1: Classification tree using full data set.

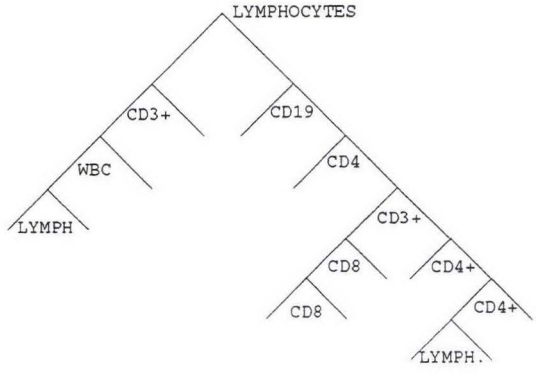


structures of the set of jackknifed trees showed great variability among each other. We found that by deleting as little as 2% of the patient data, we went from a tree splitting on 4 variables with 5 terminal nodes, to a tree splitting on 7 variables with 13 terminal nodes. This amount of variability illustrates the instability of tree models, and how little confidence we have of the insight into the problem being investigated. Figure 2 illustrates the variability in two of the trees from the jackknifed set. The tree numbers indicate which clinical trial was deleted from the data set, N is the number of observations in that trial, and the percentage is the total percent of all the cancer data that trial contributed. The variables splitting the nodes are included.

Figure 2: Two trees from jackknife set.



Tree 1 (N=42, 15% of clinical data)



Tree 9 (N=6, 2% of clinical data)

TREE STRUCTURE

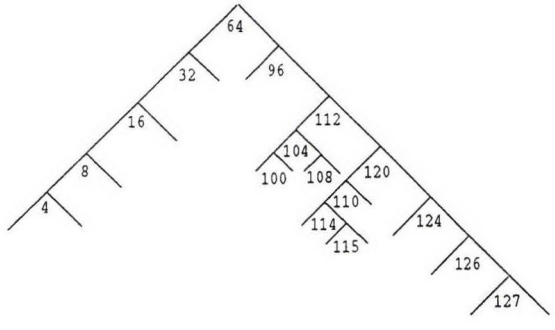
Our primary goal is to deduce, from a set of classification trees, the true tree model structure relating a set of predictor variables to known class memberships (outcomes). The true tree structure will be most generalizable among all possible trees to new data sets, and will provide the most accurate description of the mechanism relating the predictors to the outcome. To accomplish this goal we need to be able to define a distance metric between a pair of trees, and implement search algorithms for finding the tree structure with minimum distance to all other trees.

Classification trees partition a covariate space into a set of nonoverlapping hyperrectangles. Cases falling in a specific hyperrectangle are predicted to belong to the class assigned to that partition, where assignment is (usually) based on the class majority of the cases belonging to it. Selection of the hyperrectangles proceeds by recursive partitioning. Initially the entire covariate space is split into two disjoint subsets which minimize the within subset variation and maximize the between subset variation. The process is then repeated independently on the two subsets and subsequent subsets. Recursive partitioning is a greedy algorithm selecting the optimal split at each step, and does not guarantee to fit the optimal final partition.

Classification trees have a binary tree structure (Cormen, Leiserson et al. 1990) consisting of nodes and edges. Each node in a tree defines a subset of the data, has a unique identifying key, and a label defining the predictor variable forming the binary split of the data in that node into the left and right child (subset) nodes. Nodes are classified as being internal (nodes which are split into children nodes) and external or terminal (nodes which are not split). Terminal nodes are assigned a class label, and all observations falling into a specific terminal are predicted to belong to that class.

The depth of a node is the number of edges from the root node to that node. The height of a tree is equal to the maximum depth found for the terminal nodes. In the jackknife set of trees, the maximum height observed was 7. A complete binary tree has binary splits at each internal node and every terminal node occurs at the same depth. For our discussion we will consider tree structures with 255 nodes ($2^7-1=127$ internal and $2^7=128$ terminal). Internal nodes will be uniquely identified with a key number from 1,...,127, where numbering is ordered to satisfy the binary search tree property (Cormen, Leiserson et al. 1990). This states that nodes in the left subtree of any node (say the root) have an identifying key number less than the roots identifying key number. Nodes in the right subtree of a node (the root) have an identifying key number greater than the roots identifying key number. Associated with each internal node will be a splitting variable if that node is partitioned, and a nonsplit label if that node is not partitioned.

Figure 3: Nodes split in any jackknife tree.



We can now formally define the tree structure as the sequence of splitting variables ordered by the unique node key numbers. With p predictor variables and a nonsplit possible at each internal node, there are 127^{p+1} possible tree structures. (For those familiar with CART methodology, we are for the present ignoring the specification of cut points. Cut points can easily be defined after the tree structure is specified.) In the cancer study we had 12 immunological predictor variables resulting in $127^{12+1} \approx 2\times10^{27}$ possible trees of height 7. In practice the number of possible trees is less than this since terminal nodes will occur at depth smaller than 7. In these cases nodes in the subtrees rooted at terminal nodes will all be nonsplit.

Figure 3 shows the node identifying key numbers and node locations in the complete binary tree where a split was found to occur in any of the 13 jackknifed trees. The root node has key 64. Nodes not shown were never split in any of the trees and so are designated nonsplit. Table 1 (last page) show which variables split which nodes for each of the 13 trees, where the rows represent

the node key number, and the columns the tree number signifying which clinical trial was removed from the data set. The entries in the table specify the variables split on at a given node in a given tree. Nodes not split in a particular tree are designated by a dot. Node numbers not included in the table were never split.

To reconstruct a tree, say tree 5, partition the root (node #64) on lymphocytes and its right hand child (node #96) on CD3+. All other nodes are labeled nonsplit and dropped from the graphical display.

DISTANCE METRIC

Let T be the finite set of possible classification trees with height h and splitting on p predictor variables. Let $t_i, t_j \in T$ be two trees from the set T. Then $d(t_i, t_j)$ denotes an arbitrary distance metric between two trees in T. Given a distance metric we can calculate a median, or central, tree structure for T which we can use as an estimate of the true tree structure. Let t^* be the median tree minimizing the total distance

$$\min_{t^*} \sum_{t_i \in T} d(t_i, t^*).$$

We are now faced with the problems of selection of the metric, $d(t_i, t_j)$, and developing search algorithms for finding t^*.

We define the distance metric between two trees, t_1 and t_2, as

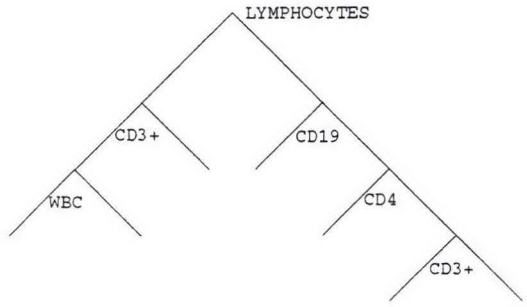
$d(t_1, t_2)$ = minimum # of nodes in t_1 added, deleted, or split on a different variable to match the structure of t_2.

This is the minimum amount of rearrangement needed in the trees to make their structures identical. The rearrangement can be an addition or deletion of a splitting node in the tree, or a change of the variable splitting a specific node. $d(t_i, t_j)$ satisfies the requirements of a metric mapping $T \times T \to R$ for $\forall t_i, t_j, t_k \in T$ since $d(t_i, t_j) \geq 0$, $d(t_i, t_j) = 0$ if and only if t_i is identical to t_j, $d(t_i, t_j) = d(t_j, t_i)$, and $d(t_i, t_j) \leq d(t_i, t_k) + d(t_k, t_j)$.

We are investigating other possible metrics for classification trees. A reasonable metric might be to penalize switching highly correlated variables less severely than switching uncorrelated variables. For interested readers, metrics on graphical structures have been discussed previously in the statistical and mathematical literature and are reviewed and applied in recent work (Margush 1982; Barthelemy and Guenoche 1991; Banks and Carley 1994; Banks and Constantine 1996).

To illustrate addition or deletion of a node refer to Figures 2 and 3. Inserting a node splitting on CD19 between tree 1's root (node #64) and its right child (node #96) increases the similarity of these two trees along the first four splits of the right hand subtree. Similarly, deleting from tree 9 node #96 (the roots' right child node) and replacing it with tree 9's node #112 splitting on CD4 produces the same increase in similarity. This addition or deletion adds 1 to the total

Figure 4: Consensus tree.



distance between the two trees, regardless of which strategy we take. The geometric result of this rearrangement is to recursively partition all observations with high absolute lymphocyte counts (the first partition) on CD4 (tree 1), or to partition only those observations having high lymphocytes and CD19 counts (tree 9).

The geometric result of changing the splitting variable in a node (not shown) is to reorient the partition of a node onto a different axis of the covariate space.

Because of the instability of classification trees it does not seem that these rearrangements are unreasonable. In fact, these can often be the result of small changes in the data set. This may be due to the presence of several good splits at a given node, the selection of one over the other being the result of fluctuations in data points (Breiman, Friedman et al. 1984).

We estimated the median tree, t^*, in the cancer study using a consensus rule constructed by identifying for each node in a tree of height 7 the variable split on most often at that node in the set of 13 jackknifed trees. From Table 1 we can calculate the consensus tree directly by selecting the variable split on most often at each node (row of the table). The consensus tree structure, shown in Figure 4, has two more splits in the left subtree of the root (nodes #32 and #16) than was found in the tree fit using all the data (Figure 1). It is easy to calculate $d(t_{\text{All Data}}, t_{\text{Consensus}}) = 2$ by deleting the two nodes in this left subtree.

The consensus tree is more centrally located than the tree using all the data as seen by comparing the total summed distances among the entire set of jackknifed trees, T_m, with the tree using all the data,

$$\sum_{t_i \in T_m} d(t_i, t_{\text{All Data}}) = 49,$$

and the tree using the consensus structure,

$$\sum_{t_i \in T_m} d(t_i, t_{\text{Consensus}}) = 39.$$

Additionally, a complete–linkage, hierarchical clustering dendogram, Figure 5, shows the consensus tree appears more centrally located than the tree using all the data.

We are currently developing software to search for the median tree using a steepest ascent algorithm.

STATISTICAL CONSIDERATIONS

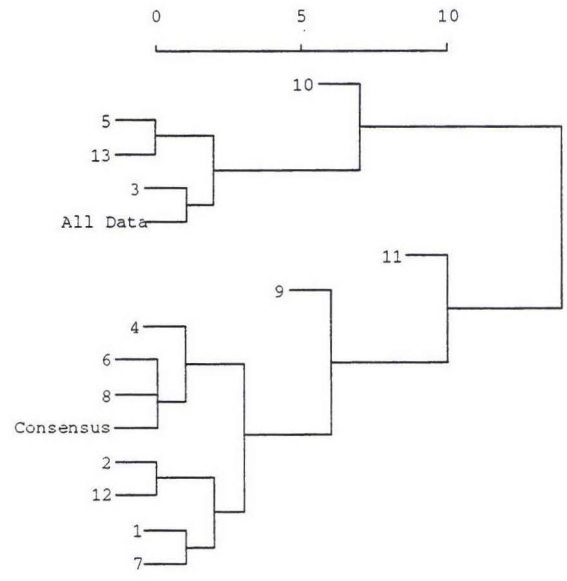
We would prefer to have had access to independent trees for this problem, for example, having fit classification trees independently to each clinical trial. However, several of the studies had very small samples sizes which prevented individual analyses. Using a

jackknife approach seemed to be an appropriate compromise, especially in light of the fact that removal of as little as 2% of the clinical data produced significant changes in tree structure.

We are investigating a family of probability models for the equivalence class of CART graphical models (i.e., those that ignore the value of the cut points but retain tree structure), mimicking work previously done (Mallows 1957; Banks and Carley 1994; Banks and Constantine 1996). This will allow us to develop maximum likelihood estimators of the central tree, goodness–of–fit tests, confidence regions and hypothesis testing frameworks for classification trees.

We are also investigating the statistical and performance properties of the median tree as a classifier, measured in terms of its learning and test set misclassification rate. Simulation studies will also allow us to measure bias and consistency properties of the median tree.

Figure 5: Dendogram.



SIGNIFICANCE

Stabilization of classification tree structure is important for uncovering the mechanism connecting the predictor variables with outcomes. Previous attempts have been focused on reducing the classification error, and are appropriate when accurate prediction is of primary importance. Our method provides a new way to stabilize this type of model while retaining the interpretability of a single classification tree.

Additionally, this method could have important applications in the analysis of very large databases where multiple trees are generated from small random samples and combined into a final model, as well as in meta–analyses were several independent studies produce classification trees using the same predictor and outcome variables. With sampling and resampling approaches, and new methods for combining graphical models, we should be able to tackle larger and more complex data analysis problems. We hope this work will stimulate further research into this area.

REFERENCES

Banks, D. and K. Carley (1994). "Metric inference for social networks." Journal of Classification **11**: 121-149.

Banks, D. and G. Constantine (1996). "Maximum entropy models for graph-valued random objects." Mathematical Social Sciences **Submitted**.

Barthelemy, J. and A. Guenoche (1991). Trees and Proximity Representations. New York, Wiley.

Breiman, L. (1992). "Stacked regressions." UC Berkeley Statistical Technical Reports **367**.

Breiman, L. (1994). "Bagging predictors." UC Berkeley Statistical Technical Reports **421**.

Breiman, L. (1994). "Heuristics of instability and stabilization in model selection." UC Berkeley Statistical Technical Reports **416**.

Breiman, L., J. Friedman, et al. (1984). Classification and regression trees. Monterey, Wadsworth and Brooks.

Clark, L. and D. Pregibon (1992). Tree-Based Models. Statistical Models in S. J. Chambers and T. Hastie. Pacific grove, Wadsworth and Brooks.

Cormen, T., C. Leiserson, et al. (1990). Algorithms. Cambridge, The MIT Press.

Langley, P. (1996). Elements of Machine Learning. San Francisco, Morgan Kaufman.

Mallows, C. (1957). "Non-null ranking models I." Biometrika **44**: 114-130.

Margush, T. (1982). "Distances between trees." Discrete Applied Mathematics **4**: 281-290.

Quinlan, J. (1993). C4.5: Programs for Machine Learning. San Mateo, CA., Morgan Kaufman.

Schurman, J. (1996). Pattern classification: A unified view of statistical and neural approaches. New York, John Wiley and Sons, Inc.

Wolpert, D. (1992). "Stacked generalization." Neural Networks **5**: 241-259.

Table 1: Structures of the jackknife tree set.

	Tree Number												
NODE	1	2	3	4	5	6	7	8	9	10	11	12	13
4	.	.	.	.	.	.	.	.	.	.	CD8	.	.
8	.	.	.	.	.	.	.	.	LYM	.	LYM	.	.
16	WBC	WBC	.	WBC	.	WBC	WBC	WBC	WBC	.	WBC	WBC	.
32	CD3+	CD3+		CD3+	.	CD3+	CD3+	CD3+	CD3+	.	CD3+	CD3+	.
64	LYM	LYM	LYM	LYM	LYM	LYM	LYM	LYM	LYM	LYM	LYM	LYM	LYM
96	CD4	CD19	CD19	CD19	CD3+	CD19	CD19	CD19	CD19	CD19	CD4	CD19	CD19
100	.	.	.	.	.	.	.	.	.	.	DR	.	.
104	CD8	.	.	.	.	.	.	.	.	CD3	CD3+	.	.
108	.	.	.	.	.	.	.	.	.	CD4+	.	.	.
112	CD3+	CD4	CD4	CD4	.	CD4	CD4	CD4	CD4	CD3+	CD19	CD4	CD4
114	.	CD8	.	.	.	.	.	.	CD8	.	CD8	CD8	.
115	.	.	.	.	.	.	.	.	.	.	CD8	.	.
116	.	CD8	.	.	.	.	CD8	.	CD9	.	WBC	CD8	.
120	.	CD3+	.	CD3+	.	CD3+	CD3+	CD3+	CD3+	CD4+	CD3+	CD3+	.
124	.	.	.	CD3+	.	.	.	.	CD4+	CD8+	CD3+	.	.
125	.	.	.	.	.	.	.	.	LYM	.	.	.	.
126	.	.	.	.	.	.	.	.	CD4+	CD19	.	.	.
127	.	.	.	.	.	.	.	.	.	CD8	.	.	.

An Incremental Construction of a Nonparametric Regression Model

Jan Smid
Department. of Mathematics
Morgan State University
Baltimore, MD 21239
smid@gsfc.nasa.gov

Petr Volf
UTIA
The Czech Academy of Sciences
Prague 8, CZ 182 08
volf@utia.cas.cz

Section: Stat. Strategy
AIStats97, Ft.Lauderdale, Fl, Jan 4-7, 97

Abstract

We study incremental (adaptive) algorithms for nonparametric regression models (one -hidden layer neural networks). Models are constructed from sets of localized functional units (e.g. radial basis functions). Both growing and pruning of a model is controlled by statistical tests and penalized error measures. The models are grown/pruned using different statistical/information criteria. Our numerical experiments show that the t-test and information type of criteria for growing/prunning for an incremental model perform at about the same level of effectivness.

1 Introduction

1.1 The Relevance to AI/Engineering Community

This paper deals with the problem of modeling of an unknown functional dependence between input and output variables of a system. The functional dependence modeling is highly relevant as a computer-aided decision tool in monitoring streams of data characterizing an instrument. For example, the Hubble Space Telescope (HST) batteries need to be monitored to assure their health and to detect a potential failure or degradation of the batteries. Functional dependencies parameters and their change in time can be incorporated into existing HST visual inspection tools. This tool can give the HST engineering deeper insight into the complex behavior of the batteries and the entire spacecraft system.

1.2 Existing Methods

One of the best known methods for fitting functions that includes interactions is Friedman's MARS, [3], and a similar method due to Breiman, [2]. MARS does not use localized units. Basis functions are added incrementally during learning, using the technique of sequential forward selection, which can be viewed as a tree-like technique, (Bishop [1]).

We deal in this paper with a class of incremental methods that use localized basis functions. Localized units offer more flexibility. For example, Weierstarss theorem guarantees approximation of any continuous function but other basis of localized functions are more effective in applications. Other drawback of global units is their inability to represent an abrupt change of the model structure. Incremental algorithms typically add a new basis function (unit) to the set of old units and then a set of parameters (one or more), typically in a lower dimensional parameter space, is adjusted.

Incremental algorithms perform better when we need to learn new features and keep the old ones as well, or if we only need to slightly modify the existing model. Incremental methods also allow us to exploit some abstract theorems and algorithms and the heuristic based on them. The advantage of an incremental procedure shows also when we deal with semi-data-driven model. What we mean by semi-data-driven model is the situation where some structural information, but not complete information, about a system is known ahead of time. When the structural information is missing we need to approach the data modeling

incrementally and locally. Data-driven modeling and model-driven-modeling are extreme cases of the semi-data-driven modeling. In the model driven approach we assume a definitive functional structure of the model and look for unknown parameters. In the data-driven approach we assume no functional structure that is to be determined from the data.

1.3 Functional Dependence Modeling

The estimator is built from a set of localized basis functional units, e.g. B-splines or radial basis functions (RBF). The presence of random noise casts the problem in the context of statistical nonparametric regression. We propose a procedure for both growing and pruning of the estimator. The growing is based on incremental fitting of residuals. We deal with a number of stochastic and deterministic procedures. The presented methods are generalized for the case of likelihood models (e.g. GAM [4], logistic models for classification, Cox's nonparametric model). The selection of units and stopping rules employ criteria of statistical analysis, namely the criteria of the least penalized residual squares. There is a number of choices among these criteria (AIC, BIC, variants of cross–validation). Our numerical experiments and data case studies have been based on the BIC (Bayes Information Criterion) and the gamma criterion. We have also used standard methods of statistical inference, namely the t-test, and ANOVA (Analysis of Variance), respectively.

In the case of a multivariate input we use the additive model and models with interactions both based on univariate and multivariate units. The additive model allows us to use the same penalization methods and it also reduces complexity of the estimator.

Consider variables $\boldsymbol{X}$ and Y as the input (predictor) and output (response) of a system that are respectively p and 1 dimensional. i.e. $\boldsymbol{X}$ takes on values $\mathbf{x}$ from R^p, and Y takes on values y from R^1. We first assume that the data $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$, $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots . x_{ip})$, are mutually independent realizations of a pair of random variables $(\boldsymbol{X}, Y)$, and are generated by a nonparametric regression function r

$$y = r(\boldsymbol{x}) + e$$

where e stands for random noise with mean zero and a constant variance σ^2. Such a general model applies not only to standard regression problems, but also to recursive cases, including general autoregression, e.g. $y_t = r(\mathbf{x}_t) + e_t$, $y_t = r(y_{t-1}, y_{t-2}, \ldots . . , y_{t-k}) + e_t$, respectively.

Our goal is to construct a function $\hat{r}(\boldsymbol{x})$ that approximates a regression function $y = r(\boldsymbol{x})$. We shall consider $\hat{r}$ constructed from localized basis functions. Such an estimator can also be viewed as a feedforward neural network with one hidden layer constructed form units (basis functions). There are two problems that need to be solved in this approach: First, how to choose units (basis functions) and of their number. Second, how to avoid data overfitting. In many cases the number of input variables (predictors) is unknown and also needs to be optimized.

Each procedure of the model construction has to answer three questions:

(i) How to select the unit which is the best candidate for innovation?

(ii) How to decide whether the innovation is effective?

(iii) When to stop the procedure?

2 Procedure of solution

We will consider units (localized basis functions) selected from a set P or a sequence of sets $\{P_j\}$ of units. There is a number of reasonable choices for a pool of units. For example a pool of units is a set of all radial basis functions $P_j = span\left\{B\left(\frac{\|\mathbf{x}-\mathbf{c}_i\|}{b}\right), b \in R;\ \mathbf{x}, \mathbf{c}_i \in R^d, i = 1, 2, \ldots, I; a \text{ constant } \alpha_o\right\}$ with the widths b and centers $\mathbf{c}$. The dimension of $\mathbf{x}$ can change, i.e. $d = d(i)$ is a function. Other choice of a set P is the additive model based on univariate units, and yet other one is the model with interactions, i.e. we consider products of univariate units.

2.1 Selecting Nonlinear Units

After we determined the pool of units a selection process is defined for an incremental process. We can make selection of a unit stochastically or we can optimize units' parameters deterministically. As a model

situation we will consider a selection process of centers of radial basis functions. The basic strategy is to select a unit and an external parameter(s) that decrease RSS. The iteration is defined as

$$r_m = \beta_{m-1} r_{m-1} + \alpha_m B_m(\mathbf{c}_i)$$

or if we do want to modify the old parameters non-uniformly

$$r_m = r_{m-1} + \alpha_m B_m(\mathbf{c}_i),\ r_{m-1} = \sum_{j=1}^{m-1} \beta_{(m-1)j} B_j(\boldsymbol{x})$$

where r_m is the model after the m-th iteration, α, β are external parameters. This selection can be substantiated by Jones' theorem. Jones' theorem, [6, 7, 8], can be paraphrased as follows: iterations

$$r_m = (1 - \alpha_m) r_{m-1} + \alpha_m B_m(\mathbf{c}_m),\ \text{known as convex algorithm}$$

$$r_m = \beta_m r_{m-1} + \alpha_m B_m(\mathbf{c}_m), \text{known as linear algorithm}$$

converge to a target function $r(\mathbf{x}) : I^d \rightarrow R$, provided that B_m, α_m, β_m and $\mathbf{c}_m$ are chosen in an almost optimal way. To avoid nonlinear optimization problems we choose new units based on following heuristic rules.

1. Random selection of centers between existing adjacent centers, or at the center of gravity of a cluster of d-dimensional points.
2. Bisecting an interval given by existing adjacent centers (a deterministic variant of the random selection).

Remark 1 *The above procedures can be generalized for units defined by a set of some parameters , not necessarily in terms of centers and widths. An example of other set of parameters are knots of (B)splines.*

As the rule of thumb for the selection of the unit's diameter we set the width value to the distance of the center of a candidate unit to the center of the nearest neighboring unit.

2.2 Finding External Coefficients

Let us assume that $\{B_j(\mathbf{x}),\ j = 1, \ldots, m-1\}$ is a set of basis functions (old units), $B_m(\mathbf{x})$ is a new unit, from which the estimator is constructed. The estimator will then have the form of a linear combination $\hat{r}(\mathbf{x}) = \alpha_o + \sum_{j=1}^{m} \alpha_j B_j(\mathbf{x})$. Since we do not need to solve nonlinear problem the optimal values of external parameters $\hat{\boldsymbol{\alpha}} = (\boldsymbol{\alpha_o}, \boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}, \ldots, \boldsymbol{\alpha_m})$ can be directly found

1. by solving the linear least squares problem

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \sum_{\mathbf{i=1}}^{\mathbf{m}} (\mathbf{y_i} - \hat{\mathbf{r}}(\mathbf{x_i}))^{\mathbf{2}}.$$

2. or for each candidate B_m we solve a one dimensional least-squares problem to obtain a coefficient α_m. This approach defines a model incrementally and we can apply the following criteria for growing and pruning. This method has clear heuristic and is computationally very convenient.

2.3 Criteria for Model Growing

Let the residuals $e_i = y_i - \hat{r}_{m-1}(\boldsymbol{x}_i)$, correspond to an estimator

$$\hat{r}_{m-1}(\boldsymbol{x}) = \sum_{j=1}^{m-1} \alpha_j\ B_j(\boldsymbol{x}).$$

Now the data $(\boldsymbol{x}_i, e_i)$ is to be fitted by a new candidate unit, selected by one of the methods described above. Let choose the most successful method of our numerical experiments. For each candidate B_m, we solve a one dimensional least-squares problem to obtain a coefficient α_m.

$$\hat{r}_m(\boldsymbol{x}) = \hat{r}_{m-1}(\boldsymbol{x}) + \alpha_m \, B_m(\boldsymbol{x})$$

Now, the penalized error criterion

$$PE = \ln(MSE) + \text{complexity term}$$

determines, whether a new unit is effective. We have used several criteria, namely BIC of Schwarz with the complexity term $m \ln(n)/n$ and the γ–criterion, with the complexity term $mn^{-\gamma}$, where γ is selected from $(0.5, 1)$. They have different strengths and we can thus choose the more or less conservative strategy.

2.4 Use of the t–statistics for Model Growing and Pruning

In conventional statistics the t–test has been used for linear regression models to show whether certain linear coefficient can be set to zero without significant increase of the residual variance. The same procedure can be used for testing of external coefficients α of nonlinear units. In the growing phase, we accept a new unit B_m if the t–test indicates that α_m differs from zero significantly. In the pruning phase, we can perform the t–test for all external parameters (or for their subset) simultaneously. We simply delete that unit - i.e. we set to zero that coefficient - for which a t–statistics value lies below a critical value for the t–test (with $n - m - 1$ degrees of freedom, m is a number of units in the model, n is the number of data points). Simultaneously the penalizing criteria (e.g. gamma or BIC) are used as the additional indicators with the t–test.

2.5 Iteration of optimal additive model

Let us assume in this paragraph that basis functions $B(x_k)$ have one dimensional input x_k. By an additive model, we mean that (Breiman [2])

$$r(\boldsymbol{x}) = \sum_{k=1}^{p} g_k(x_k), \quad \mathbf{x} = (x_1, x_2, ...x_p)$$

Each function $g_k(x)$ can be modeled the sum of basis functions. The model then reads in terms of basis functions

$$r(\boldsymbol{x}) = \alpha_0 + \sum_{k=1}^{p} \sum_{j=1}^{m_k} \alpha_{jk} \, B_{jk}(x_k).$$

It is clear that growing and pruning of the additive model can be handled by the same procedure as the model with one dimensional input.

2.6 Models containing interactions of predictors

Let us now consider an estimator in the form of linear combination of units with multi-dimensional input vector. We assume that the units with multi-dimensional input are the tensor products of univariate unit functions. We follow C. J. Stone's "dimensionality reduction principle" [5] and keep the dimensionality of the model as low as possible. This means that we prefer models containing none or low–dimensional interactions. If necessary we choose the maximal dimension of interactions in advance and increase it during an adaptive process .

The procedure of model growing and pruning is essentially the same as in the case of strictly additive structure. The iterative selection of new basis functions runs through domains of components $x_1, x_2, ..., x_p$, then through all domains of couples of components (x_j, x_k), and so on.

3 Likelihood Models

In such models, the response function r , is a parameter of the conditional distribution of Y, given $\boldsymbol{X} = \boldsymbol{x}$. Let us consider a model $y = r(\boldsymbol{x}; \boldsymbol{\alpha}) + \varepsilon$ for non-Gaussian noise, $r(\boldsymbol{x}; \boldsymbol{\alpha}) = \sum_{j=1}^{m} \alpha_j \, B_j(\boldsymbol{x})$. Denote by $f(y; r(\boldsymbol{x}; \boldsymbol{\alpha}))$ the probability density of Y for an unknown parameter $\boldsymbol{\alpha}$. Then,

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \ln f(y_i; r(\boldsymbol{x}_i; \boldsymbol{\alpha})),\ \boldsymbol{\alpha} = (\alpha_1, \alpha_{2,...,}\alpha_m)$$

is the logarithm of likelihood function. The inference in such models is based on the maximum likelihood principle (i.e. to find an optimal vector $\boldsymbol{\alpha}$ we have to solve nonlinear equations $\partial\mathcal{L}/\partial\alpha_j = 0,\ j = 1,...,m$). In order to reduce the complexity of computations we propose incremental procedures for growing and pruning steps. Our numerical experiments showed that a several Newton–Raphson iterations suffice to solve one new parameter α_i of a nonlinear likelihood equation based on localized units.

3.1 Criteria of acceptance of a new model.

For Gaussian noise, the logarithm of likelihood is proportional to the averaged sum of squares of residuals, i.e. to the estimate of residual variance. As we have seen, this estimate, suitably penalized by model complexity, may serve as a criterion of acceptance. In the case of a likelihood model, we can employ the penalized log-likelihood in order to decide whether the update of a model is effective. We can also find an analogy of the t–tests for selecting the candidates for pruning. The maximum likelihood estimator has, under certain regularity conditions, favorable asymptotic properties, namely the estimator is consistent and asymptotically normally distributed. The variance of this distribution is given by the negative inverse of the matrix of the second derivatives of the log–likelihood. Therefore, we can compute approximately normal variables for the test of hypothesis and set coefficients α_j to zero correspondingly.

4 Applications

Using the methods proposed in this paper we have solved a number of examples, namely standard Gaussian nonparametric regression problems, nonparametric Cox's model of the hazard rate and logistic classification model. Numerical results of our experiments are encouraging. In this paper we present two case studies, a test problem and a real-world data problem. The first one deals with the approximation of a noisy sinusoidal function. A real-world problem deals with the analysis of the voltage and the current of Hubble Space Telescope nickel-hydrogen batteries.

4.1 Noisy Sinusoidal Example:

We have generated 200 points x_i distributed randomly uniformly over the interval $(0, 5)$. A function $y_i = x_i \sin(x_i^2) + e_i$ has been sampled (Fig.1). The components of Gaussian noise e_i were sampled independently from the normal distribution $N(\mu = 0, \sigma = 0.5)$. Our goal was to reconstruct, from the data $(x_i, y_i),\ i = 1,..,200$, a function $r(x) = x\sin(x^2)$. Our algorithm employed Gausian radial basis functions, namely $B(x) = \frac{1}{\sqrt{2\pi}b}\exp(-\frac{(x-c)^2}{2b^2})$. The centers c were optimized by the BIC and Gamma penalization procedures, while the radii b were adapted automatically, namely set to the distance of the center of a candidate unit to the center of the nearest neighboring unit.

The procedure was initialized by 5 equidistantly located units. The candidates for the new units were generated randomly (i.e. following the method 1). After the growing phase, the model contained 15 units (Fig.5). The growing phase was stopped when 3 consequent global iterations did not change the model. We used the minimum of the BIC and Gamma (with $\gamma = 0.8$) criteria. Both criteria behave very similarly. The MSE of the model was $\hat{\sigma}^2 = 0.32666$.

Then we apply the pruning procedure, based on the t–test choice of units for deletion. This phase ended up with the final model of 10 units (Fig.3) The t–test criterion was compared with the penalizing criteria. There were no significant differences in performance between these three criteria. Final MSE was $\hat{\sigma}^2 = 0.37302$.

4.2 Hubble Space Telescope Nickel-Hydrogen Batteries Modelling

In a real–data case, we modeled the dependence of the voltage (Fig.2) on the current (Fig.6), during repeated periods of charging and discharging. We considered the additive model of a three component function $V(t) = r^*(C, t, d) + e(t)$, where $r^*(C, t, d) = r_1(C(t)) + r_2(C(t-d)) + r_3(C(t-2d))$, t denotes

the time, d is the time lag constant. One dimensional functions r_j were modeled by RBF. The process of model growing started with 5 RBF units for each component, after 50 global iteration loops, the model had 10, 16, 18 units, respectively, the MSE was 0.0424. This phase was controlled by the gamma criterion, $\gamma = 0.7$. Then, the pruning phase, controlled by the t-tests, reduced the model to 6, 4, 10 units (Fig.4) and reached the final MSE = 0.0465.

5 Conclusion

We have experimented with a several incremental methods of building a nonparametric regression model and several growing and stopping criteria. The incremental method using localized units has some advantages over the batch modeling methods.

- It reduces computational complexity by allowing to solve a sequence of low dimensional optimization problems that are easier to handle than a high dimensional nonlinear problem.
- It allows us to implement heuristically based iteration that decrease RSS and avoid nonlinear optimization.
- It is a good starting point for solving optimal modeling problems with respect to other measures of deviation than RSS. For instance, for exponential families of models or logistic classification models rigorous estimation of linear parameters $\boldsymbol{\alpha}$ requires an iterative procedure. Therefore, each possible reduction of computations is desirable. For more discussion about this approach, see also Buja et al. in [3].
- Our studies indicated that the use of different growing/pruning strategies, namely the t-statistics criterion and penalization criteria provided approximately same results. The advantage of the t-test is in its theoretical justification.

References

[1] Bishop Ch.(1995): *Neural Networks for Pattern Recognition. Clarendon* Press, Oxford.

[2] Breiman L. (1993): Fitting additive models to data. *Computational Statistics and Data Analysis* 15, 13–46.

[3] Friedman J.H. (1991): Multivariate adaptive regression splines; with discussion. *Annals of Statistics*, vol. 19, pp. 1–141.

[4] Hastie T. and Tibshirani R. (1993): *Generalized Additive Models.* Chapmann and Hall, London.

[5] Stone C.J. (1994): The use of polynomial splines and their tensor products in multivariate function estimation. *Annals of Statistics* 22, pp. 118–194.

[6] Jones, L. K.,(1992): A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Annals of Statistics*, vol.20, pp. 608-613.

[7] Kurkova V. and Smid J.(1995): An incremental architecture algorithm for feedforward neural nets. *Preprints of the European IEEE Workshop Computer Intensive Methods in Control and Signal Processing*, Prague, 1994.

[8] Kurkova, V:Personal communication.

[9] Barron A.R. (1993): Universal approximation bounds for superposition of a sigmoidal function. *The IEEE Transactions on Information Theory*, vol. 39, 930-945, 1993.

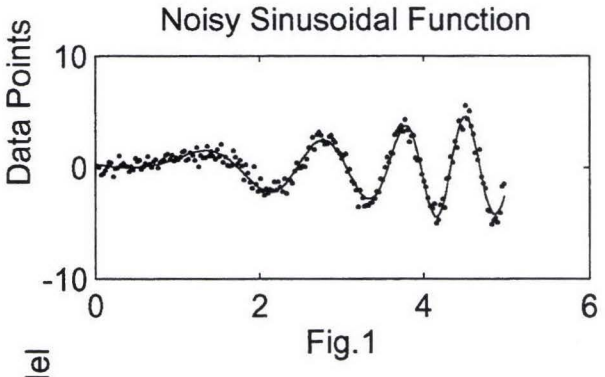


Fig.1

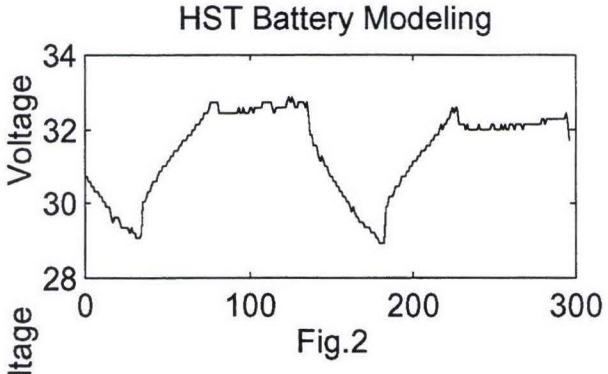


Fig.2

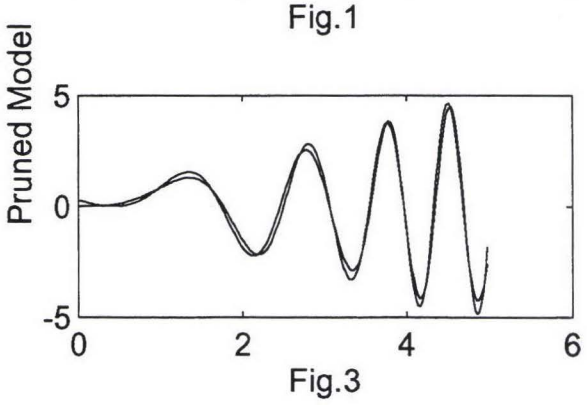


Fig.3

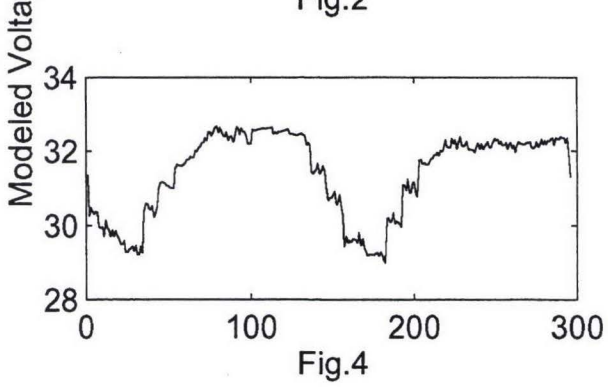


Fig.4

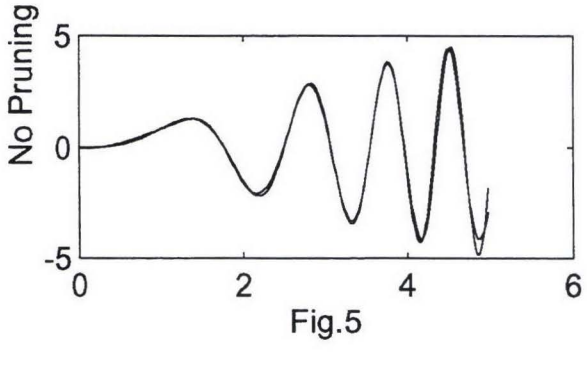


Fig.5

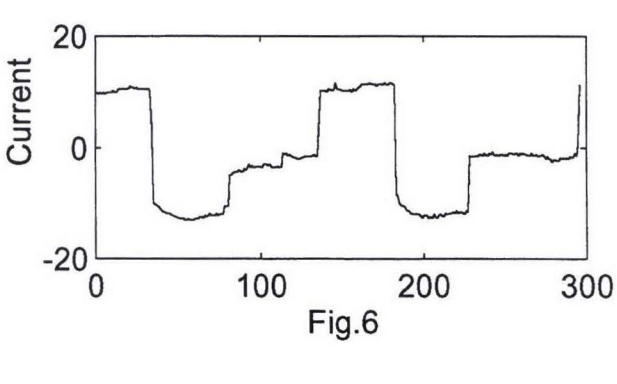


Fig.6