Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Adaptive Hinging Hyperplanes

Jun Xu * **Xiaolin Huang** * **Shuning Wang** *

* *Tsinghua National Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing, 100084,CHINA (e-mail: yun-xu05@mails.tsinghua.edu.cn)*

Abstract: The model of adaptive hinging hyperplanes (AHH) is proposed in this paper for black-box modeling. It is based on Multivariate Adaptive Regression Splines (MARS) and Generalized Hinging Hyperplanes (GHH) and shares attractive properties of the two. By making a modification to the basis function of MARS, AHH shows linear property in each subarea. It is proved that AHH model is identical to a special case of the Generalized Hinging Hyperplanes (GHH) model, which has a universal representation capability for continuous piecewise linear functions. AHH algorithm is developed similar to MARS algorithm. It is adaptive and can be executed quickly, hence has power and flexibility to model unknown relationships. In addition, due to the piecewise-linear property, AHH is preferred to MARS when modeling high-dimensional dynamic systems, especially when the sample size is small and under noise conditions.

Keywords: Systems and signals; black-box modeling; adaptive regression; piecewise-linear

1. INTRODUCTION

In the field of black-box modeling, the basis function expansion is generally used, i.e., the unknown relationships are approximated by functions of linear combinations of basis functions (Friedman [1994]),

$$\hat{f}(\mathbf{x}) = \sum_{k=0}^{M} a_k B_k(\mathbf{x})$$

while $\mathbf{x}$ is a vector of predictor variables, a_k is the expansion coefficient and $B_k(\mathbf{x})$ is the basis function. Different kinds of basis functions yield different models, such as projection pursuit (Friedman and Stuetzle [1981]), wavelet neural networks (Pati and Krishnaprasad [1990]), constrained topological mapping (Cherkassky and Lari-Najafi [1991]), multivariate adaptive regression splines (MARS) (Friedman [1991]), continuous piecewise-linear approximation (Wang and Sun [2005]) and so on. Among all these methods, MARS and piecewise linear approximation possess their unique advantages.

MARS is simple, fast and efficient, which is due to its adaptiveness and identification method: least-squares. It combines the idea of recursive partitioning regression (CART) (Breiman et al. [1984]) with function representation based on tensor-product splines. The advantages of the two methods: adaptive adjusting and continuity, are preserved during the combination. The MARS model can be interpreted as a tree where each node in the tree consists of a basis function and uses a tree-based algorithm for constructing the model. The nodes are split according to a goodness of fit measure, but unlike other recursive partitioning methods, all nodes (not just the leaves) are candidates for splitting. Only the least-squares is used during each splitting, so the execution speed is very fast. The basis function for MARS is a product of univariate splines each with a directional term, which is so called truncated power splines, hence, combinations of these basis functions are continuous. A forward stepwise strategy searching splitting nodes as well as a backward stepwise strategy removing leaves is used in the MARS procedure. Simulation results in Friedman [1991] showed that MARS applied well in nonlinear system identification, not only effective for additive functions but also for high-dimensional functions with variables interacted with each other.

In the MARS model, the order of the basis function is increased with the dimension of the problem. In general, high-order splines are unstable and more likely to provide misleading structures, which is more serious in the case of noise and small sample size. Unfortunately, real dynamic system is often under noise and the sample data is always not easy to obtain, therefore, the application of MARS in high-dimensional dynamic system is restricted.

The continuous piecewise-linear model is linear in each subarea, hence represent more stability than high-order splines. There are several piecewise linear models, such as Canonical Piecewise Approximation (Kahlert and Chua [1990]), Lattice Piecewise-linear Representation (Tarela et al. [1990]), Hinging Hyperplanes model (Breiman [1993]), among which the last is used widely for regression, classification and function approximation. The basis function of this model is as follows

$$\max\{0, l(\mathbf{x})\} \tag{1}$$

where $l(\mathbf{x})$ is a linear function. As (1) shows, 2 hyperplanes given by 0 and $l(\mathbf{x})$ are joined together at $\{\mathbf{x} : l(\mathbf{x}) = 0\}$, and (1) is also called the hinge (function). The model can approximate a large class of nonlinear functions to arbitrary precision as long as it contains sufficient hinges. However, the HH model cannot represent all the continuous piecewise linear functions (CPWL) in more than 2 dimensions. In Wang and Sun [2005], by adding a certain number of linear functions in (1), Wang introduced the Generalized Hinging Hyperplanes (GHH) model and proved the representation ability of the model in any dimensions. There have been some approximation algorithms since the model was proposed, such as Sun and Wang [2005],Wen et al. [2007], all of which are based on Hinge Finding Algorithm (HFA) (Breiman [1993]). Actually, HFA is a Newton algorithm applied to a sum

of squared error criterion (Pucar and Sjöberg [1998]). Hence, even the convergence of the algorithm to a local minimum cannot be guaranteed. Due to no rapid and effective algorithm exists for parameter identification of GHH, the applications of it is restricted.

Inspired by MARS (Friedman [1991]), in this paper, we introduce the model of adaptive hinging hyperplanes (AHH), the basis function of which is obtained by a small modification of that of MARS. It is proved that AHH is actually one kind of GHH with restrictions imposed on the division of the domain. Like the MARS algorithm, only least-squares is needed in the AHH algorithm, hence no nonlinear parameters exist in this algorithm and it runs very fast. From computational view, it is much superior compared to HFA and the series of algorithms mentioned above as those algorithms are actually Newton algorithm and infected by the convergency problem. Besides, the AHH model has advantages over the MARS model in high dimensional dynamic system identification due to the stableness of linear functions.

The paper is organized as follows. Section 2 gives a brief review of MARS and GHH, which are the basis of AHH. Then, in Section 3, the AHH model and algorithm are introduced. Section 4 compares the application of MARS and AHH in high dimensional dynamic system, taking the influence of noise and small sample size into account. Simulations are done in section 5, including approximating 2-dimensional functions using HH, AHH and MARS, and comparing AHH with MARS in a 5-dimensional nonlinear dynamic system.

2. A BRIEF REVIEW OF MARS AND GHH

2.1 MARS

MARS (Multivariable Adaptive Regression Splines) was first introduced by J. H. Friedman (Friedman [1991]). The method is presented for flexible regression modeling of high dimensional data. It can be considered as a generalization of recursive partitioning regression, which is generally viewed as a geometrical procedure. The idea is to partitioning the entire domain recursively, assigning a constant value for each subregion. In short, the procedure is to cast the approximation to the original function in the form below:

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} a_m B_m(\mathbf{x}) \tag{2}$$

The basis function $B_m(\mathbf{x})$ takes the form

$$B_m(\mathbf{x}) = I[\mathbf{x} \in R_m] \tag{3}$$

where I is an indicator function having the value one if its argument is true and zero otherwise. The regions are chosen based on a greedy optimization procedure where in each step the algorithm selects the split which causes the largest decrease in mean squared error. There are problems existed in recursive partitioning regression, such as discontinuous among subregions and unable to provide good approximation to certain classes of simple often-occurring functions. These are functions that either have no interaction effects, or strong interactions each involving at most a few of the predictor variables.

To overcome the above problems, the MARS procedure uses continuous basis functions rather than the indicator functions. The basis functions of MARS can be expressed in terms of the product of truncated power spline function $[\pm(x_\upsilon - t)]_+^q$, where the subscript $[]_+$ denotes the positive part of the expression, q is the order of univariate spline function, x_υ is the split variable and t is the knot location. Besides, MARS differs from recursive partitioning regression in that it makes all the subregions eligible for further splitting, while in the latter only newborn subregions at the previous step is availble. This makes it powerful to model relationships both additive and involving interactions in at most a few variables.

The MARS algorithm is comprised of a forward stepwise and backward stepwise strategy. In the forward stepwise procedure, a search is performed over all subregions to find the next subregions, after a certain number of splits, the backward stepwise procedure incorporate subregions (remove basis functions) that no longer contribute sufficiently to the accuracy of the fit. The model generated following the MARS procedure ($q = 1$) is of the form

$$\hat{f}(\mathbf{x}) = a_1 + \sum_{m=2}^{M} a_m \prod_{k=1}^{K_m} \left[s_k^m \cdot (x_k^m - t_k^m) \right]_+ \tag{4}$$

where $a_1, \ldots, a_M$ are the coefficients of the basis functions, a_1 is for the constant basis function. The sum is over the basis functions B_m produced by the forward stepwise procedure that survive the backwards deletion strategy. K_m is the number of splits that gave rise to B_m, $s_k^m = \pm 1$, x_k^m and t_k^m denote the split variable and split point at the k-th split for the m-th basis function. Always restricting the splitting area to the original domain gives rise to additive MARS model, i.e., $K_m = 1$

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} a_m \hat{g}_m(x_m) \tag{5}$$

The measure of fit used by the MARS algorithm is a modified form of the generalized cross validation (GCV) estimate (Craven and Wahba [1979]):

$$\mathrm{LOF}(\hat{f}_M) = \frac{1}{N} \sum_{i=1}^{N} \left[y_i - \hat{f}_M(\mathbf{x}_i) \right]^2 \Big/ \left[1 - \frac{\tilde{C}(M)}{N} \right]^2 \tag{6}$$

whereas $\tilde{C}(M)$ is called the cost complexity function and can be expressed as

$$\tilde{C}(M) = \mathrm{trace}(\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T) + 1 + d \cdot M \tag{7}$$

where $\mathbf{B}$ is the $M \times N$ data matrix of the M (nonconstant) basis functions. The quantity d is a smoothing parameter of the procedure, representing the cost for each basis function optimization. Larger values for d will lead to fewer knots being placed and thereby smoother function estimates. It provides an estimate of the future prediction accuracy by measuring the mean squared error on the training set and penalizing this measurement to account for the increase of variance due to model complexity.

2.2 GHH

The model of generalized hinging hyperplanes (GHH) was first introduced by Wang and Sun [2005] as a generalization or extension of the model of hinging hyperplanes (HH), which was first raised by Breiman (Breiman [1993]). The HH model is a sum of hinge functions like

$$H(\mathbf{x}) = \sum_{m=1}^{M} \delta_m h_m(\mathbf{x})$$

where $\delta_m = \pm 1$ and the hinge function $h_m(\mathbf{x})$ is given by (1), $h_m(\mathbf{x}) = \max\{0, l_m(\mathbf{x})\}$. Linear function $l_m(\mathbf{x})$ can be expressed as $l_m(\mathbf{x}) = [1, \mathbf{x}^T]\theta_m$ with $\mathbf{x} \in R^n$ and θ_m is an $n+1$ dimensional coefficient vector.

It is proved by Breiman that if $f(\mathbf{x})$ is a sufficiently smooth function, there is a constant $C(f)$ such that for any positive integer M, there are hinge functions $h_1, \ldots, h_M$ with

$$\left\| f(\mathbf{x}) - \sum_{m=1}^{M} h_m(\mathbf{x}) \right\| \leq \frac{C}{M}$$

The HH model can approximate a large class of nonlinear functions to arbitrary precision. However, they can represent only a small part of continuous piecewise-linear (CPWL) functions in more than 2 dimensions. In Wang and Sun [2005], it is proved that any CPWL function $p(\mathbf{x})$ of n variables can be represented by a sum of generalized hinges containing at most $n+1$ linear functions, i.e.

$$\sum_m \delta_m \hat{h}_m(\mathbf{x}) \tag{8}$$

where the generalized hinge function is of the form

$$\hat{h}_m(\mathbf{x}) = \max\left\{0, [1\ \mathbf{x}^T]^T\theta_{m1}, \ldots, [1\ \mathbf{x}^T]^T\theta_{mk_m}\right\}$$

with $\delta_m = \pm 1$ and $k_m \leq n$. (8) is called n-order hinging hyperplanes, or generalized hinging hyperplanes (GHH). As GHH covers more continuous piecewise-linear functions, it is more flexible than the HH model for black-box modeling.

3. ADAPTIVE HINGING HYPERPLANES

3.1 AHH model

In MARS, the truncated power spline function with $q = 1$ is $[\pm(x_\upsilon - t)]_+$, in fact, it can be written as $\max\{0, \pm(x_\upsilon - t)\}$, which is actually a hinge function of 2-order. Therefore, additive MARS model is actually equivalent to one kind of the HH model, for which the boundary of each subregion is parallel to the axis. Since additive MARS model has some relations with hinge functions, would it be possible to obtain generalized hinge functions based on nonadditive MARS model?

The answer is positive. The model of adaptive hinging hyperplanes is obtained by a slight modification to the basis function of MARS, i.e., the operator "×" is replaced by "min". Hence, the basis function of AHH is

$$\min_{k\in\{1,\ldots,K_m\}}\{\max\{0, s_k^m \cdot (x_k^m - t_k^m)\}\}$$

where K_m is the number of univariate truncated power spline function (hinge function) contained in a basis function and items in the "min" bracket must involve distinct predictor variables. It is proved that AHH is equivalent to one special kind of the GHH model.

Theorem 1. Let

$$B_m(\mathbf{x}) = \min_{k\in\{1,\ldots,K_m\}}\{\max\{0, s_k^m \cdot (x_k^m - t_k^m)\}\} \tag{9}$$

be the m-th basis function of AHH, then the following identity is valid for all $\mathbf{x} \in R^n$

$$\begin{aligned} B_m(\mathbf{x}) = {} & \min\{s_1^m(x_1^m - t_1^m), \ldots, s_{K_m}^m(x_{K_m}^m - t_{K_m}^m)\} \\ & - \min\{0, s_1^m(x_1^m - t_1^m), \ldots, s_{K_m}^m(x_{K_m}^m - t_{K_m}^m)\} \end{aligned} \tag{10}$$

Proof. Let I_m equals the set $\{i | s_i^m(x_i^m - t_i^m) > 0\}$, the proof is continued according to the value of I_m.
(i) $I_m = \emptyset$
$B_m(\mathbf{x})$ equals 0 and the right-hand side of (10) becomes $s_{j_0}^m(x_{j_0}^m - t_{j_0}^m) - s_{j_0}^m(x_{j_0}^m - t_{j_0}^m) = 0$, where $s_{j_0}^m(x_{j_0}^m - t_{j_0}^m)$ is the minimum of $s_i^m(x_i^m - t_i^m), (i = 1, \ldots, K_m)$. (10) holds.
(ii)$I_m = \{1, \ldots, K_m\}$
$B_m(\mathbf{x})$ equals to

$$\min\{s_1^m(x_1^m - t_1^m), \ldots, s_{K_m}^m(x_{K_m}^m - t_{K_m}^m)\}$$

The right-hand side of (10) becomes
$\min\{s_1^m(x_1^m - t_1^m), \ldots, s_{K_m}^m(x_{K_m}^m - t_{K_m}^m)\} - 0$
$= \min\{s_1^m(x_1^m - t_1^m), \ldots, s_{K_m}^m(x_{K_m}^m - t_{K_m}^m)\}$
(10) holds.
(iii)$I_m \subseteq \{1, \ldots, K_m\}$ but not equal and $I_m \neq \emptyset$ and $s_{i_0}^m(x_{i_0}^m - t_{i_0}^m)$ is the least among all the variables.
$B_m(\mathbf{x})$ is 0 at this time and the right-hand side of (10) becomes

$$s_{i_0}^m(x_{i_0}^m - t_{i_0}^m) - s_{i_0}^m(x_{i_0}^m - t_{i_0}^m) = 0$$

(10) still holds.
Therefore, the identity holds for all $\mathbf{x} \in R^n$.

Corollary 2. The result of AHH procedure

$$\hat{f}(\mathbf{x}) = a_1 + \sum_{m=2}^{M} a_m B_m(\mathbf{x}) \tag{11}$$

with $B_m(\mathbf{x})$ being as the form of (9), is equivalent to one special kind of the Generalized Hinging Hyperplanes model.

Proof. From Theorem 1, (11) is equivalent to

$$\begin{aligned} a_1 \ & + a_2 \min\{s_1^2(x_1^2 - t_1^2), \ldots, s_{K_2}^2(x_{K_2}^2 - t_{K_2}^2)\} \\ & - a_2 \min\{0, s_1^2(x_1^2 - t_1^2), \ldots, s_{K_2}^2(x_{K_2}^2 - t_{K_2}^2)\} \\ & + \cdots \\ & + a_M \min\{s_1^M(x_1^M - t_1^M), \ldots, s_{K_M}^M(x_{K_M}^M - t_{K_M}^2)\} \\ & - a_M \min\{0, s_1^M(x_1^M - t_1^M), \ldots, s_{K_M}^M(x_{K_M}^M - t_{K_M}^2)\} \end{aligned}$$

As each basis function is restricted to involve distinct predictor variables in the "min" bracket, at most n distinct items are included in each "min" bracket.

Compared to the Generalized Hinging Hyperplanes model

$$\sum_m \delta_m \max\left\{[1\ x^T]^T\theta_{m1}, \ldots, [1\ x^T]^T\theta_{mk_m}\right\}, k_m \leq n+1$$

AHH model is actually a special kind of GHH model with linear functions in the "max" (actually "-min") bracket specified with $s_j(x_j - t_j), j \in \{1, \ldots, n\}$. Consequently, the boundaries of subregions are those satisfying $x_j = t_j$ or $s_j(x_j - t_j) = s_k(x_k - t_k), j, k = 1, \ldots, n$.

3.2 AHH algorithms

Analog to the MARS algorithm, the AHH forward and backward algorithms are developed. The GCV criterion is also used. Use $\upsilon(k, m)$ to denote the variable of the k-th factor in the m-th basis function, the algorithm is as follows.

Algorithm 1(AHH-forward stepwise)

1. Set the first basis function $B_1(\mathbf{x}) = 100$ (or some positive integer large enough to make $\min\{B_1(\mathbf{x}), B_m(\mathbf{x})\} = B_m(\mathbf{x})$ when

$m > 1$), initial number of the basis functions $M = 1$.
2. Choose a basis function $B_m(\mathbf{x}), (m = 1, \ldots, M)$, a splitting variable $\upsilon \notin \{\upsilon(k,m)|1 \le k \le K_m\}$ and a splitting point t. Let
$g = \sum_{i=1}^{M} a_i B_i(x) + a_{M+1} \min\{B_m(x), \max\{x_\upsilon - t, 0\}\}$
$+ a_{M+2} \min\{B_m(x), \max\{-(x_\upsilon - t), 0\}\}$
use least-squares to obtain an optimal approximation of g to the data with respect to coefficients $a_1, \ldots, a_{M+1}, a_{M+2}$.
3. Choose $B_m^*(\mathbf{x})$, corresponding splitting variable x_υ^* and knot t^* as those that yield the best fit, then

$$B_{M+1}(x) = \min\{B_m^*(x), \max\{x_\upsilon^* - t^*, 0\}\}$$
$$B_{M+2}(x) = \min\{B_m^*(x), \max\{-(x_\upsilon^* - t^*), 0\}\}$$

4. If $M < M_{\max}$, go to 2. Else, exit.

As the backward stepwise is concerned, the model containing $M_{\max}$ basis functions obtained above is considered as the initial model. At each iteration, one basis function is to be deleted, which is the one whose removal either improves the fit the most or degrades it the least. Hence, a sequence of $M_{\max} - 1$ models is constructed, each one having one less basis function than the previous one in the sequence. Choose the model fit the best as the final model.

4. COMPARISON OF MARS AND AHH IN HIGH-DIMENSIONAL NONLINEAR DYNAMIC SYSTEM

Consider the nonlinear dynamic system

$$y(t) = g(\varphi(t), \theta) + \varepsilon(t) \tag{12}$$

where $\varphi(t) = \varphi(u^{t-1}, y^{t-1})$ is the regression vector, $y(t) \in R$ is the measured output and $\varepsilon(t) \in R$ is the error term.

When the regression vector $\varphi(t)$ consists of previous inputs and outputs

$$\varphi(t) = [1, y(t-1), \ldots, y(t-n_a), u(t-1), \ldots, u(t-n_b)]^T$$

the system is defined as an NARX system (Sjöberg et al. [1995]).

Given an adequate data set without noise, both MARS and AHH exhibit good approximation ability, or may to some extent, MARS even perform better. However, in a real NARX system, some unfavorable conditions exist. First, sample data available is limited. Then, output noise affects. Last, as the regression vector contains previous outputs, input uncertainty appears. All the 3 factors affects the prediction to a certain extent.

Assume $\mathbf{x} = \phi(t)$, according to Friedman [1997], the random nature of the training data T implies that the estimate $\hat{g}(\mathbf{x}|T)$ is a random variable. Assume the output noise ε is a white noise, therefore, the expected prediction error can be expressed as

$$\begin{aligned} & E_T[y - \hat{g}(\mathbf{x}|T)]^2 \\ &= E_T[g(\mathbf{x}) - \hat{g}(\mathbf{x}|T) + \varepsilon]^2 \\ &= E_T[g(\mathbf{x}) - \hat{g}(\mathbf{x}|T)]^2 \\ &= [g(\mathbf{x}) - E_T\hat{g}(\mathbf{x}|T)]^2 + E_T[\hat{g}(\mathbf{x}|T) - E_T\hat{g}(\mathbf{x}|T)]^2 \end{aligned} \tag{13}$$

The term $E_T[y - \hat{g}(\mathbf{x}|T)]^2$ represents the squared prediction error (at $\mathbf{x}$) averaged over repeatedly realized training samples of the (same) size N from the system under study. The term $E_T[g(\mathbf{x}) - \hat{g}(\mathbf{x}|T)]^2$ is the squared "estimation error" in the target function $g(\mathbf{x})$ averaged over training samples. From the last equality of (13), we can see it depends only on the mean and the variance of the distribution of $\hat{g}(\mathbf{x}|T)$. The bias $g(\mathbf{x}) - E_T\hat{g}(\mathbf{x})$ represents how closely on average the estimate is able to approximate the target. The variance $E_T[\hat{g}(\mathbf{x}|T) - E_T\hat{g}(\mathbf{x}|T)]^2$ reflects the sensitivity of the function estimate $\hat{g}(\mathbf{x}|T)$ to the training sample T. More sensitivity means that the estimates will be more variable under sampling variations in the data.

Generally speaking, the bias and variance increases with decreasing training sample size. Due to the polynomial structure of the MARS procedure, it tends to give a more precise description of the training data but larger bias and variance when predicting. From (13), this larger bias and variance lead to larger prediction error at $\mathbf{x}$. Hence, when the sample size is small , the MARS procedure may provide a misleading indication of the association between the response and the predictor variables.

On the other hand, the output noise $\varepsilon(t)$ affects the identification by adding random ingredients to the training data T, both on the input and output. It is easy to see that linear structure is more stable when the output was added a random element. Following gives a brief analysis of how input uncertainties affects the identification procedure in the MARS and AHH models.

In MARS, the products of truncated power spline function $[\pm(x-t)]^q$ is included in each basis function, as the dimension increases, the order of the basis function increases. On the other hand, AHH takes the minimum of several truncated power spline functions, hence the order remains to be 1 in any dimensions. Suppose in n-dimensional $\mathbf{x}$-space, the input uncertainties for predictor variables (actually noise on the output) are $\delta x_1, \cdots, \delta x_n$, and consider changes of the two kinds of basis functions in the corresponding valid subregions. In (4), assume the basis function is of the form $B_m(\mathbf{x}) = \prod_{k=1}^{n} [s_k \cdot (x_k - t_k)]_+$, for the case of simplicity, suppose each s_k equals 1, then when measurement uncertainty exists, the relative change of $B_m(\mathbf{x})$ is

$$\begin{aligned} \frac{\Delta B_m(\mathbf{x})}{B_m(\mathbf{x})} = & \frac{\delta x_1}{x_1 - t_1} + \cdots + \frac{\delta x_n}{x_n - t_n} \\ & + \frac{\delta x_1 \delta x_2}{(x_1 - t_1)(x_2 - t_2)} + \cdots + \frac{\delta x_{n-1} \delta x_n}{(x_{n-1} - t_{n-1})(x_n - t_n)} \\ & + \cdots + \frac{\delta x_1 \cdots \delta x_n}{(x_1 - t_1) \cdots (x_n - t_n)} \end{aligned}$$

In case of (11), analog to the assumptions made above, the change of $B_m(\mathbf{x}) = \min\{x_1 - t_1, \cdots, x_n - t_n\}$ is

$$\frac{x_i - t_i + \delta x_i - (x_j - t_j)}{x_j - t_j} < \frac{\delta x_j}{x_j - t_j}$$

in which $x_j - t_j$ is the least when the measurement uncertainty is not considered and $x_i - t_i + \delta x_i$ is the least afterwards.

As the coefficient matrix $\mathbf{B}$ for the least-squares is comprised of the value of basis functions at each sample point, from the above two expressions, we can expect AHH procedure give more stable identification results when measurement uncertainty affects.

Overall, under the 3 factors mentioned in the begining of this section, the AHH procedure provides a more stable identification. This directly yields less bias and variance of the prediction $\hat{g}(\mathbf{x}|T)$, thus smaller prediction error. Therefore, the AHH model is more suitable than the MARS model when the NARX system is considered and the superiority would become more significant as the dimension grows.

5. SIMULATION EXAMPLES

In the following simulation examples, we present the result of applying the AHH procedure to 2 examples. It is shown that when modeling nonlinear phenomenon, AHH is accurate, flexible and computationally effective. In the examples below, when MARS is used to compare the effects with those of AHH, continuous but not smooth piecewise polynomial splines is used. d in the GCV criterion (6) is set to be 2(Friedman [1991]).

In section 5.1, the approximation of a 2-dimensional function is handled using a special HH model, AHH model and MARS model. Section 5.2 applies AHH and MARS to a 5-dimensional dynamic system with white noise and confirms results obtained in section 4.

5.1 Modeling 2-Dimensional Function

Consider the following nonlinear function

$$y = f(\mathbf{x}) = e^{-\delta_1\|\mathbf{x}-0.3\mathbf{e}\|_2^2} - e^{-\delta_2\|\mathbf{x}-0.7\mathbf{e}\|_2^2} \tag{14}$$

where $\mathbf{e} = [\,1\ 1\,]^T$. The example is first studied in Breiman [1993] to inspect the approximating ability of the HH model, here, we use it to show the approaching capability of the AHH model and compare it to that of the HH model with the domain partitioned parallel to the axis (equivalent to additive MARS model). It is realized by placing constraints on the parameter *mi* in the AHH model, which is the maximum number of variables allowed to appear in any basis function, $K_m \le mi$. In this example, $mi = 1$ corresponds to the special HH model, and $mi = 2$ is equivalent to the AHH model. The MARS procedure is also applied. There are 500 points sampled uniformly on the square $[0\ 1]^2$. The response is assigned according to

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad 1 \le i \le N \tag{15}$$

with ε_i randomly generated from a standard normal distribution. Here the signal-to-noise ratio is 3.28 so that the true underlying function accounts for 91.5% of the variance of the response. The training error and prediction error are all calculated by

$$E = \sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \Big/ \sum_{i=1}^{N}(y_i - \bar{y}_i)^2 \tag{16}$$

where $\hat{y}_i$ is the approximation value and $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. The prediction error is calculated over 5000 sample data regenerated. As the problem to be considered is simple (in 2-dimension), the maximum number of the basis function for each method is set to be 15. All the computation were performed with MATLAB6.5 on Pentium(R) 4. For this simple example, computation cost for all the 3 cases (HH, AHH, MARS) is small. For example, in one experiment, when $\delta_1 = \delta_2 = 16$, time consuming is 1.28s, 2.97s and 3.86s respectively. Table 1 compares the training and prediction error of applying these 3 strategies, of which TE and PE stand for the 2 errors respectively. Errors listed in Table 1 are mean values of 100 replications, also shown in Table 1 are the corresponding standard deviations (in parentheses) over the 100 replications.

Comparing the two kinds of errors of the 3 situations, AHH and MARS exhibit their superior approximating ability over HH, which is more significant when the function to be approximated becomes sharp. However, the powerful ability of HH cannot be ignored. In some situations such that the function to be approximated is smooth or in low-dimension, the HH procedure will be chosen if its error is only slightly (say 5 or 10 percent) worse due to its simpleness. Besides, the behaviors of AHH and MARS differed little in this example.

Table 1. Comparison of the accuracy of HH, AHH and MARS modeling

Errors		$\delta_1 = \delta_2 = 4$	$\delta_1 = \delta_2 = 9$	$\delta_1 = \delta_2 = 16$
TE	HH	0.0381(0.0027)	0.0734(0.0048)	0.0996(0.0082)
	AHH	0.0067(0.0014)	0.0135(0.0034)	0.0306(0.0084)
	MARS	0.0057(0.0011)	0.0175(0.0036)	0.0342(0.0080)
PE	HH	0.0499(0.0312)	0.0993(0.0265)	0.1340(0.0333)
	AHH	0.0058(0.0018)	0.0132(0.0040)	0.0344(0.0093)
	MARS	0.0046(0.0013)	0.0181(0.0039)	0.0394(0.0140)

5.2 Modeling High-Dimensional Nonlinear Dynamic System

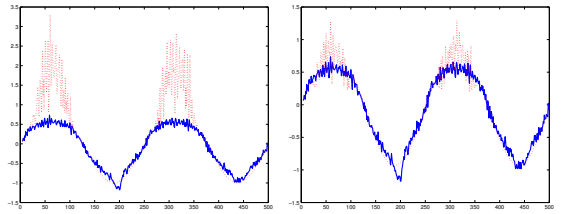
Consider the nonlinear dynamical system described by the input-output model (Narendra and Parthasarathy [1990])

$$y(t) = \frac{y(t-1)y(t-2)y(t-3)u(t-2)(y(t-3)-1)}{1+y^2(t-2)+y^2(t-3)} + \frac{u(t-1)}{1+y^2(t-2)+y^2(t-3)} \tag{17}$$

To generate the identification data, the system is excited with a random input signal $u(t)$ uniformly distributed in the interval $[-0.5, 0.5]$. To provide a complete comparison between the MARS and AHH procedure, 4 situations are considered: training set contains 200 ($1 \le t \le 200$) or 2000 points ($1 \le t \le 2000$), without or with noise (the noise is conformed to normal distribution $N(0, 0.05^2)$). 20 basis functions are allowed for both MARS and AHH. 500 points are used as the testing data, for which the input is defined as

$$u_t = \sin(2\pi t/50), 1 \le t \le 500 \tag{18}$$

Fig (1) shows the performance of MARS and AHH procedure over one experiment when there are 200 sample points with noise, computation cost for the two are 23.25s and 21.75s.



(a) True (solid) and simulated (dashed) output using MARS

(b) True (solid) and simulated (dashed) output using AHH

Fig. 1. Using MARS and AHH simulate system (17) with input (18)

We can see from fig (1) that outputs deviation generated by MARS was much greater than that of AHH in this experiment.

To explicitly observe the overall trend of the training error and prediction error, the summary consists of the percent points of the lower half of the distribution of the two errors for the MARS and AHH model. These distributions were obtained by applying MARS and AHH to 100 data sets. The percent points of the lower half of a distribution X is often called α-lower quantiles,

which is the smallest value K with $P(X \leq K) = \alpha$. For example, the value 0.0002 in the third row, second column of table 2 reflects that when using MARS, the training error was no more than 0.0002 10 times out of the 100 experiments.

Table 2. Lower quantiles for training error

	10%	50%	70%	80%	90%
N=200 (without noise)					
MARS	0.0002	0.0003	0.0003	0.0003	0.0003
AHH	0.0007	0.0009	0.0010	0.0011	0.0014
N=200 (with noise)					
MARS	0.0313	0.0369	0.0389	0.0403	0.0430
AHH	0.0328	0.0378	0.0398	0.0411	0.0442
N=2000 (without noise)					
MARS	0.0003	0.0003	0.0003	0.0003	0.0004
AHH	0.0007	0.0011	0.0012	0.0012	0.0012
N=2000 (with noise)					
MARS	0.0351	0.0365	0.0373	0.0379	0.0388
AHH	0.0355	0.0369	0.0377	0.0383	0.0389

Table 3. Lower quantiles for prediction error

	10%	50%	70%	80%	90%
N=200 (without noise)					
MARS	0.0156	0.0750	0.1553	0.2088	0.4279
AHH	0.0272	0.0518	0.0728	0.0859	0.1123
N=200 (with noise)					
MARS	0.0554	0.1900	0.5342	0.9080	6.4992
AHH	0.0653	0.1404	0.2587	0.4239	1.1402
N=2000 (without noise)					
MARS	0.0062	0.0386	0.0611	0.0862	0.1489
AHH	0.0393	0.0478	0.0506	0.0537	0.0616
N=2000 (with noise)					
MARS	0.0292	0.1153	0.1939	0.3243	0.5080
AHH	0.0323	0.0615	0.0757	0.0950	0.1539

Table 2 shows that MARS gives a little smaller training error than AHH, which coincides with the fact that splines can describe a relationship finer than linear functions. However, under noise conditions, the differences is very small, which indicates that AHH is more resistant to noise than MARS even in approximating. As the prediction error is concerned, which is shown in Table 3, AHH exhibits significant superiority to MARS, and the superiority becomes more noteworthy when the sample is of small size or with noise. In the most unfavorable situation, when the noise is added on the 200 training points, AHH outperforms MARS almost each time of the 100 replications.

Furthermore, a much smaller variation is founded in the distribution obtained by AHH for each of the 4 situations, representing that AHH is more stable and more likely to give smaller prediction error than MARS in modeling high-dimensional dynamic systems.

6. CONCLUSION

The model of adaptive hinging hyperplanes (AHH) is developed to combine the model of multivariate adaptive regression splines (MARS) and generalized hinging hyperplanes (GHH) in a way that best retains the positive aspects of the both, while being less vulnerable to their unfavorable properties. Inheriting the property of adaptiveness from MARS, AHH shows a strong approximation ability and great flexibility. As a special case of GHH, AHH preserves all advantages of linear functions, thus is more suitable than MARS to problems of high dimension and with high noise. The AHH algorithm runs fast and no nonlinear parameters exists, which is superior to the algorithms developed for GHH before.

AHH is actually bringing GHH to nonlinear modeling in an effective way. It is promising to use it in nonlinear system identification and control.

REFERENCES

L. Breiman. Hinging hyperplanes for regression, classification and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, 1993.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

V. Cherkassky and H. Lari-Najafi. Constrained topological mapping for nonparametric regression analysis. *Neural Networks*, 4:27–40, 1991.

P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31:377–403, 1979.

J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

J. H. Friedman. An overview of predictive learning and function approximation. In V. Cherkassky, J. H. Friedman and H. Wechsler, Eds, editor, *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, volume 136 of *NATO ASI Series F*. Springer-Verlag, Berlin, 1994.

J. H. Friedman. On bias, variance, 0/1—-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1: 55–77, 1997.

J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of American Statiscal Association*, 76:817–823, 1981.

C. Kahlert and L. O. Chua. A generalized canonical piecewise linear representation. *IEEE Transaction on Circuits and Systems*, 37(3):373–382, 1990.

K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Automatic Control*, 1(1):4–27, 1990.

Y. C. Pati and P. S. Krishnaprasad. Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. Technical Report 90-44, Electrical Engineering Department, University of Maryland at College park, 1990.

P. Pucar and J. Sjöberg. On the hinge-finding algorithm for hinging hyperplanes. *IEEE Transactions on Information Theory*, 44(3):3310–3319, 1998.

J. Sjöberg, Q. Zhang, et al. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31 (12):1671–1724, 1995.

X. Sun and S. Wang. A special kind of neural networks: continuous piecewise linear functions. *Lecture Notes in Computer Science*, 3496:375–379, 2005.

J. M. Tarela, E. Alonso, and M. V. Martinez. A representation method for pwl functions oriented to parallel processing. *Mathematical & Computer Modelling*, 13(10):75–83, 1990.

S. Wang and X. Sun. Generalization of hinging hyperplanes. *IEEE Transactions on Information Theory*, 12(51):4425–4431, 2005.

C. Wen, S. Wang, et al. Identification of dynamic systems using piecewise-affine basis function models. *Automatica*, 43(10): 1824–1831, 2007.