



Unification of neural and statistical modeling methods that combine inputs by linear projection

Bhavik R. Bakshi* and Utomo Utojo

Department of Chemical Engineering, The Ohio State University, Columbus, OH 43210, USA

(Received 13 January 1997; revised 13 May 1998)

Abstract

Empirical modeling methods that combine inputs by linear projection include linear methods such as, ordinary least-squares regression, partial least-squares regression, principal components regression, and nonlinear methods such as, backpropagation networks with a single hidden layer, projection pursuit regression, nonlinear partial least-squares regression, and nonlinear principal components regression. In this paper, these popular modeling techniques are unified to yield a single method called nonlinear continuum regression (NLCR). This unification is based on the insight provided by a common framework for empirical modeling methods, and is achieved by using activation functions that adapt to the measured data, a common optimization criterion for finding the projection directions, and a hierarchical training methodology that allows efficient modeling. The adaptive-shape activation functions are determined by univariate smoothing in the space of the projected input versus output. The NLCR optimization criterion contains an adjustable parameter that controls the degree of overfitting or bias of the model, and spans the continuum of methods from projection pursuit regression or backpropagation networks to nonlinear principal components regression. Consequently, NLCR results in models that are usually more general and compact than those obtained by existing methods based on linear projection, while eliminating the need for arbitrary selection of an empirical modeling method based on linear projection for a given task. The improved modeling ability of NLCR and its performance on different types of training data are illustrated by examples based on simulated and industrial data. © 1998 Published by Elsevier Science Ltd. All rights reserved.

Keywords: linear projection; unification; neural and statistical modeling

1. Introduction

Some of the most popular empirical modeling methods developed in the fields of artificial neural networks, chemometrics, and statistics, transform the inputs by projecting them on a linear hyperplane before operation of the basis function. These methods based on linear projection combine the inputs as a linear weighted sum, and include linear methods such as, ordinary least-squares regression (OLS), partial least-squares regression (PLS), principal components regression (PCR) and ridge regression (RR), and nonlinear methods such as, backpropagation networks (BPN) with a single hidden layer, projection pursuit regression (PPR), nonlinear PLS, and nonlinear PCR. These methods have been used for developing empirical models for a large variety of process operation and control tasks (Kresta *et al.*, 1991; Venkatasubramanian *et al.*, 1990; Qin and McAvoy, 1992)

and several other engineering (Kosko, 1992; Haykin, 1994), and chemometric (Martens and Naes, 1989) tasks. All methods based on linear projection relate inputs to outputs as

$$\hat{y}_k = \sum_{m=1}^M \beta_{mk} \theta_m \left(\sum_{j=1}^J \alpha_{jm} x_j \right), \quad (1)$$

where x_j ($j = 1, \dots, J$) are the input or predictor variables, α_{jm} are the projection directions, θ_m ($m = 1, \dots, M$) are the activation or basis functions, β_{mk} are the regression coefficients, and \hat{y}_k ($k = 1, \dots, K$) are the approximated outputs or response variables. The broad diversity in methods based on linear projection arises from the different types of basis functions and optimization criteria used to determine the model parameters. These decisions about the type of basis functions and optimization criteria determine the performance of each method for different empirical modeling problems. For example, restricting the basis functions to be linear as in OLS, PLS and PCR results in linear models with efficient

* Author to whom all correspondence should be addressed.

training and easier physical interpretation. Furthermore, methods such as, OLS, BPN and PPR perform best when large quantities of training data are available, whereas PLS, NLPLS, PCR and NLPCR perform well when the ratio of training data to inputs is small and when the inputs are related. Unification of methods based on linear projection can result in hybrid techniques that combine the properties of existing methods, and perform well for various types of input-output relationships, amounts of training data, and nature of correlation between the inputs, while eliminating arbitrary selection of a modeling method for a given task.

Unification of linear modeling methods has been studied by several researchers. The optimization criteria for OLS, PLS and PCR differ in their emphasis on capturing the relationship between the inputs, which increases from OLS to PLS to PCR. Based on this insight, Stone and Brooks (1990) have unified these linear modeling methods by developing a common objective function that can specialize to OLS, PLS, or PCR by selecting the appropriate value of an adjustable parameter. Since this new adjustable parameter can take any value on the continuum between OLS and PCR, this unified method is called continuum regression (CR). Similar techniques have also been developed by Lorber *et al.* (1987) while providing a theoretical framework for PLS. The CR adjustable parameter complements the effect of the number of basis functions on the degree of overfitting or bias of the empirical model, and provides additional control over the quality of the empirical model. The technique of CR has been extended to include ridge regression (Sundberg, 1993; deJong and Farebrother, 1994), and its performance for dynamic modeling has been studied by Wise and Ricker (1993).

Research on the unification of nonlinear modeling methods based on linear projection is quite limited. A commonly suggested approach for combining the features of neural and statistical methods is to determine the initial values of the BPN input edge weights as PCA or PLS projection directions, and adapting the basis function parameters and edge weights to minimize the output prediction error (Piovosio and Owens, 1986; Martin *et al.*, 1995). This approach aims to combine the ability of PCA and PLS to provide better models with limited data, with the universal approximation property of BPN. Various NLPLS approaches that combine PLS with BPN, spline, or statistical smoothers have also been developed (Holcomb and Morari, 1992; Wold, 1992; Frank, 1990). A CR-type common objective function to combine the benefits of PLS modeling with PPR is described by Haario and Taavitsainen (1994), but their approach does not emphasize the unification of methods based on linear projection, the basis functions are only of fixed shape, and BPN are not a part of the approach. Reviews on nonlinear empirical modeling methods (Barron and Barron, 1988; Sjoberg *et al.*, 1995; Frank, 1995) also do not attempt to combine the features of various methods.

This paper presents a new empirical modeling technique that unifies all linear and nonlinear methods based on linear projection. This unification is based on the insight provided by a common framework for all empirical modeling methods (Bakshi and Utojo, 1998). This framework shows that empirical modeling methods differ from each other depending on decisions about only three aspects of the model namely, nature of the input transformation, type of activation functions, and optimization criterion for determining the model parameters. This framework indicates that unification of methods that combine inputs by linear projection requires a general method for determining activation functions of any shape, a common objective function that can specialize to existing methods based on linear projection, and an efficient training methodology. The resulting method unifies OLS, PLS, PCR, BPN with one hidden layer, PPR, NLPLS, and NLPCR. This method extends the principles of CR to nonlinear modeling, and is named nonlinear continuum regression (NLCR). Activation functions of any shape in the projected input-output space are determined by univariate smoothing techniques such as, variable span smoothers, splines, Hermite polynomials, and back-propagation networks. The common objective function for determining the projection directions is similar to CR, and subsumes all methods based on linear projection by adjusting the value of a single parameter. Finally, the training methodology includes the adaptive basis functions and general optimization criterion to extract the empirical model in an efficient hierarchical manner, and specializes to popular algorithms for existing methods based on linear projection. The optimum values of the objective function parameter and number of basis functions are determined via crossvalidation.

Since NLCR subsumes all methods based on linear projection, the resulting models are at least as good, if not better, than those obtained by existing methods based on linear projection. Selecting a value of the objective function parameter equal to zero results in PPR, BPN or OLS depending on whether the basis functions are of adaptive shape, sigmoid, or linear, respectively. A value of 0.5 for the objective function parameter results in PLS or NLPLS, and a value of 1 provides PCR or NLPCR. The NLCR training methodology adjusts this parameter to straddle the continuum between PPR or BPN and NLPCR, resulting in models that are more general, and more compact than those developed by existing methods based on linear projection. This adjustable parameter controls the degree of overfitting or model generality by affecting the bias-variance trade-off to minimize the error of approximation between the empirical and actual model. The ability of basis functions to adapt to the training data further enhances the compactness and physical interpretability of the NLCR model. This paper develops NLCR for modeling with multiple inputs and a single output.

The rest of this paper is organized as follows. The common framework for neural and statistical

methods, and the challenges for unifying existing methods are introduced in Section 2. Univariate smoothing techniques for determining activation functions of any shape are described and illustrated in Section 3. The objective functions for determining the projection directions in existing linear and nonlinear methods based on linear projection, and development of the NLCR common objective function that subsumes all methods are discussed in Section 4. A common hierarchical training methodology for NLCR is developed in Section 5. For specific values of the objective function adjustable parameter, γ , this method specializes to existing algorithms such as, the NIPALS algorithm for PCR and PLS, and the PPR algorithm. Heuristic strategies for avoiding local minima and for efficient selection of the optimum value of γ are also discussed. Examples based on synthetic data and from an industrial polymerization reactor are used to illustrate the properties of NLCR in Section 6. Finally, conclusions and directions for future work are discussed in Section 7.

2. A common comparison framework for empirical modeling methods

The model determined by all empirical modeling methods may be represented as a weighted sum of basis functions as

$$\hat{y}_k = \sum_{m=1}^M \beta_{mk} \theta_m(\phi_m(\alpha; x_1, x_2, \dots, x_J)), \quad (2)$$

where α is the matrix of basis function parameters, and ϕ_m represents the input transformation. The transformed inputs are also referred to as latent variables represented as

$$z_m = \phi_m(\alpha; x_1, x_2, \dots, x_J).$$

The terms input and output space refer to the respective spaces in which the input and output variables lie, while the term, transformed input–output space refers to the space of the latent variables and the output. The model given by equation (2) may also be represented as an artificial neural network where α are the edge weights of the input and hidden layers, β are the edge weights of the output layer, and ϕ and θ are the basis functions. Specific empirical modeling methods may be derived from equation (2) depending on decisions about the nature of input transformation, type of activation or basis functions, and optimization criteria. These decisions form the basis of the common framework for comparing all empirical modeling methods (Bakshi and Utojo, 1998), and are introduced briefly in the rest of this section. Equation (2) represents a model between multiple inputs and a single or multiple outputs. If separate models are developed for each output, the basis functions and projection directions may be different for each output. Otherwise, the basis functions and projection directions may be the same for each output. This paper focuses on modeling of multi-input–single-output systems.

2.1. Nature of input transformation

The complexity of the modeling task, and the quantity of training data required for an acceptable model quality increase significantly with the number of input variables. Empirical modeling techniques fight this “curse of dimensionality” by transforming the inputs to *latent* variables that capture the input–output relationship with less latent variables than the number of inputs. Such dimensionality reduction is usually accomplished by exploiting the relationship among inputs, or distribution of training data in the input space, or relevance of input variables for predicting the output. Thus, empirical modeling methods may be divided into three categories depending on the nature of input transformation. *Methods based on linear projection* exploit the linear relationship among inputs by projecting them on a linear hyperplane before applying the basis function. This class of methods is unified by the nonlinear continuum regression approach presented in this paper. *Methods based on nonlinear projection* exploit the nonlinear relationship between the inputs by projecting them on a nonlinear hypersurface resulting in latent variables that are nonlinear functions of the inputs. If the inputs are projected on a localized hypersurface such as a hypersphere or hyperellipse, then the basis functions are local. Otherwise, the basis functions are non-local in nature. *Partition-based methods* fight the curse of dimensionality by selecting input variables that are most relevant to efficient empirical modeling. The input space is partitioned by hyperplanes that are perpendicular to at least one of the input axes.

2.2. Type of activation functions

The wide variety of activation functions used in empirical modeling methods may be broadly divided into two categories depending on whether their shape is fixed or adaptive. The activation function, $\theta(z)$, relates the transformed input, z , to the output, and is two-dimensional in nature. The shape of the activation function and type of the input transformation determine the nature of the basis function, $\theta(\alpha; x)$, which relates the inputs to the output. The three-dimensional basis function, and its corresponding activation function are shown in Fig. 1. For this example, the inputs are transformed by linear projection. *Fixed-shape* activation functions such as, linear, sigmoid, Gaussian, wavelet, or sinusoid are commonly used in several modeling methods. Adjusting the basis function parameters changes their location, size, and orientation, but their shape is decided *a priori*, and remains fixed. Some empirical modeling methods relax the fixed-shape requirement and allow the basis functions to adapt their shape, in addition to their location, size, and orientation, to the training and testing data. This additional degree of freedom provides greater flexibility in determining the unknown input–output surface, and often results in more compact models. *Adaptive-shape* basis functions

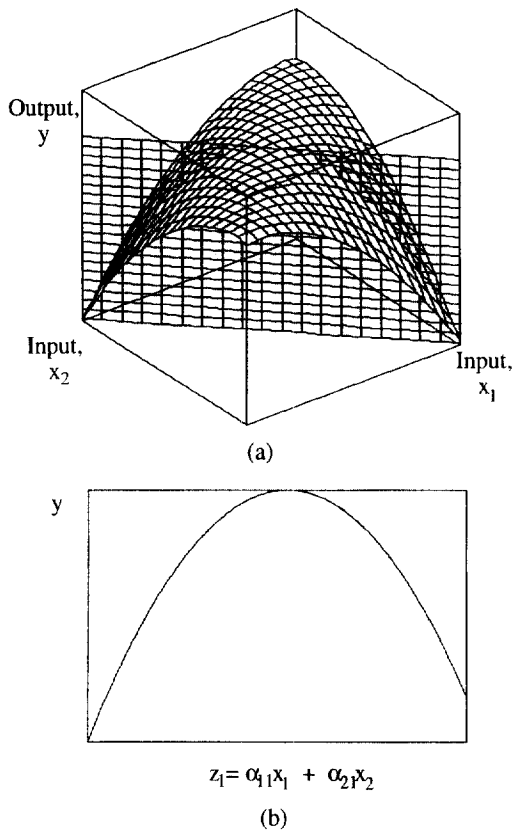


Fig. 1. Relationship between basis function and activation function. (a) Basis function relates all the inputs to the output. (b) Activation function corresponding to the basis function in (a), relates the transformed inputs to the output.

are obtained through the application of smoothing techniques such as, splines, variable span smoothers, and polynomials to approximate the transformed input–output space.

2.3. Optimization criteria

The input transformation is determined by the function, ϕ , and parameters, α , whereas the model relating the transformed inputs to the output is determined by the parameters, β , and basis functions, θ . Empirical modeling methods often use different objective functions for estimating the parameters that determine the input transformation, and those determining the relationship between the transformed input and output. This separation of the empirical modeling optimization criteria permits explicit control over the dimensionality reduction by input transformation, and may result in more accurate empirical models as demonstrated in Section 6. Empirical modeling methods may be divided into two categories depending on whether the optimization criterion for the input transformation contains information from only the inputs, as in maximizing the variance of the measured data captured by the transformed inputs by PCA, or both inputs and outputs, as in maximizing

the covariance between the transformed inputs and outputs by PLS. The optimization criterion for determining the output parameters and activation functions is to minimize the output prediction error, and is common to all empirical modeling methods.

The nature of the input transformation, type of basis functions, and optimization criteria discussed in this section provide a common framework for comparing the wide variety of techniques for input transformation and input–output modeling, as depicted in Table 1. This comparison framework is useful for understanding the similarities and differences between various methods, and may be used for selecting the best method for a given task, and to identify the challenges for combining the properties of various techniques. Thus, unification of methods based on linear projection requires techniques for determining activation functions that can specialize to any fixed or adaptive shape, and a common optimization criterion for determining the input transformation parameters that can specialize to various existing methods depending on the nature of the modeling problem. Finally, an efficient training methodology that uses the general activation functions and optimization criterion is essential. The remainder of this paper describes techniques for meeting these challenges leading to the unification of empirical modeling methods based on linear projection. This class of methods is given by equation (1), and the latent variable is a weighted sum of the input variables.

3. Techniques for determining adaptive activation functions

Unification of the variety of basis functions used in methods based on linear projection requires a general activation function that can assume any linear or nonlinear shape depending on the nature of the training data. Such activation functions may be obtained by using univariate smoothing techniques for approximating the training data in the projected input–output space. A variety of techniques are available for determining the general activation functions including, variable span smoothers (Friedman, 1984), Hermite functions (Hwang *et al.*, 1994), automatic smoothing splines (Roosen and Hastie, 1994), and backpropagation networks. Any technique for determining general activation functions needs to possess the following characteristics:

- ability to smooth arbitrarily spaced data containing different amounts and types of noise and curvature,
- fast implementation with minimum storage requirements,
- easy computation of the activation function values for testing data and its derivatives.

The smoothing problem is inherently ill-posed, and all techniques need to determine the appropriate degree of smoothness that provides the best fit. The characteristics of various techniques for determining

Table 1. Comparison matrix for empirical modeling methods (Bakshi and Utojo, 1998)

Method	Input transformation	Basis function	Optimization criteria
OLS	Linear projection	Fixed shape, linear	α — Max. squared correlation between projected inputs and output β — Min. output prediction error
PLS	Linear projection	Fixed shape, linear	α — max. covariance between projected inputs and output β — Min. output prediction error
PCR	Linear projection	Fixed shape, linear	α — Max. variance of projected inputs β — Min. output prediction error
BPN single	Linear projection	Fixed shape, sigmoid	$[\alpha, \beta]$ — Min. output prediction error
PPR	Linear projection	Adaptive shape, supersmoother	$[\alpha, \beta, \theta]$ — Min. output prediction error
BPN mult.	Nonlinear projection, nonlocal	Fixed shape, sigmoid	$[z, \beta]$ — Min. output prediction error
NLPCA	Nonlinear projection, nonlocal	Adaptive shape	$[\alpha, \phi]$ — Min. input prediction error
RBFN	Nonlinear projection, local	Fixed shape, radial	$[\sigma, \tau]$ — Min. distance between inputs and cluster center β — Min. output prediction error
CART	Input partition	Adaptive shape, piecewise constant	$[\beta, \tau]$ — Min. output prediction error
MARS	Input partition	Adaptive shape, spline	$[\beta, \tau]$ — Min. output prediction error

the general activation functions and their use in empirical modeling are discussed in the rest of this section.

Variable span smoothers. A simple smoothing technique is to fit a line to data in a window of a selected length or span. If the curvature and noise in the data change over time, it is essential to vary the span of the window to vary the degree of smoothing in an optimum manner. The supersmoother (Friedman, 1984) is such a nonparametric variable span smoother. The data, $\{(z_1, y_1), (z_2, y_2), \dots, (z_L, y_L)\}$, in a selected window of length L are approximated by a least-squares fit of a straight line given by

$$\theta(z_i) = b + az_i \frac{-L}{2} \leq i \leq \frac{L}{2}, \quad (3)$$

where, z_i are the values of the linearly projected inputs. The parameters, a and b in equation (3) are computed for different values of the span, and the best span for each data point is selected via cross-validation. Friedman (1984) suggests span values of $L = 0.05I, 0.2I,$ and $0.5I$ corresponding to high, medium, and low frequency, respectively. The smoothness of the curve obtained by the crossvalidated span may be further improved by smoothing once again to enhance lower frequencies specified by a user-defined tone control parameter. Additional details of this algorithm are described by Friedman (1984).

The supersmoother adapts well to all types of smoothness and noise, and is a fast algorithm due to the simple calculations. Unfortunately, the activation function is defined only at the values of the training

data. Consequently, determining the activation function values at points other than the training data requires interpolation or extrapolation based on assumptions of the smoothness between the adjacent training data. This necessitates storage of the activation function values at all the training data, resulting in increased storage requirements. Furthermore, other properties of the activation functions such as, its derivatives, must be estimated by direct numerical differentiation, which may degrade the performance of the supersmoother activation functions for the previously unseen data.

Hermite polynomials. For the given data, (z_i, y_i) , the smoothed function may be obtained by fitting Hermite functions of order Q as,

$$\theta(z_i) = \sum_{q=1}^Q c_q h_q(z_i), \quad (4)$$

where c_q are the regression coefficients and $h_q(z)$ are the orthonormal Hermite functions expressed as

$$h_q(z) = (q!)^{-1/2} \pi^{1/4} 2^{-(q-2)/2} H_q(z) \psi(z)$$

with $H_q(z)$ and $\psi(z)$ being the orthogonal Hermite polynomials and the Gaussian function, respectively. The Hermite polynomials, $H_q(z)$, can be constructed in a recursive manner as

$$H_0(z) = 1,$$

$$H_1(z) = 2z,$$

$$H_q(z) = 2(zH_{q-1}(z) - (q-1)H_{q-2}(z)).$$

The regression coefficients, c_q , in equation (4) are computed by the pseudo-inverse to minimize the least-squares error of approximation.

Using Hermite functions for univariate smoothing overcomes some of the drawbacks of the supersmoothen (Hwang *et al.*, 1994). Since Hermite functions may be computed analytically and recursively, neither storage of the smoothed function nor piecewise interpolation are required for prediction with testing data. Other properties such as the derivative may be easily computed analytically, instead of numerically as in the supersmoothen. The order or number of Hermite functions used in the smoothing may be determined by crossvalidation or may be specified by the user.

Smoothing Splines. A smoothing spline is fit to minimize the penalized least-squares criterion given by

$$\sum_{i=1}^I \{y_i - \theta(z_i)\}^2 + \lambda \int \{\theta''(t)\}^2 dt, \quad (5)$$

where, the second term in equation (5) is the roughness penalty which penalizes for large curvature via the parameter λ . The value of λ controls the smoothness of the function, with a larger value of λ resulting in a smoother fit. The value of the smoothing parameter may be chosen automatically by generalized crossvalidation (Roosen and Hastie, 1994). Approaches for overcoming the tendency of this criterion to undersmooth short trends as high-frequency structure have also been developed. Automatic smoothing splines do not require storage of the smoothed function to compute values for testing data, and the basis function derivatives may be computed from the spline functions.

Backpropagation networks. BPN with fixed-shape basis functions may also be used for determining the univariate basis functions. The smoothness of the fit is determined by the number of hidden nodes in the BPN, which may be selected by crossvalidation with testing data. Predicting basis function values for testing data requires storage of all the BPN parameters which may be less compact than Hermite polynomials or cubic splines for some basis function shapes. BPNs may also not provide the best approximation of the data in case of convergence to local minima.

Illustrative example. The performance of the smoothing techniques for determining the activation functions is illustrated by considering the following function,

$$y = \sin((2\pi(1 - z)^2) + z\varepsilon), \quad (6)$$

where ε is independent and identically distributed Gaussian noise with zero mean and unit variance. The curvature of equation (6) decreases and the variance of the random component increases with increasing abscissa value. This function was used by Friedman (1984) to illustrate the behavior of the supersmoothen. The data consist of 200 pairs (z, y) with z drawn randomly from a uniform distribution in $[0, 1]$, and y given by equation (6).

Performance of the supersmoothen for three values of the tone control parameter, 0, 5, and 10, is shown in Fig. 2a. Increasing the value of the tone control parameter results in a smoother curve. A tone control parameter value of 5 is recommended as a reasonable choice for damping out some of the high frequency variation (Friedman, 1984). The resulting smooth curve captures the curvature and variation of the data reasonably well. The performance of Hermite polynomials of different orders is shown in Fig. 2b. Comparing the result of all methods in Fig. 2c illustrates that for the proper degree of smoothness, the performance of each smoothing techniques is quite similar since each method can capture the curvature and variation in the data, and produce reasonably smooth curves. More exhaustive comparison of the supersmoothen with Hermite polynomials is presented by Hwang *et al.* (1993) and of the supersmoothen with automatic spline smootheners by Roosen and Hastie (1994). The examples solved in Section 6 are based on the supersmoothen.

4. General optimization criterion for projection directions

Methods based on linear projection differ in the optimization criterion used for determining the projection directions, as shown in Table 1. Consequently, their unification requires a general optimization criterion that consists of information from both the inputs and output, and can specialize to the criterion used by existing methods based on linear projection. Such a general optimization criterion for multi-input, single-output modeling by the linear methods of OLS, PLS and PCR has been proposed by Stone and Brooks (1990) as

$$\max_{z_m} \{[\text{corr}^2(\mathbf{y}, \mathbf{X}z_m)] [\text{var}(\mathbf{X}z_m)]^\gamma\}, \quad (7)$$

where, \mathbf{y} is the vector of output measurements. The objective function given by equation (7) specializes to OLS, PLS, and PCR for γ equal to 0, 1, and ∞ , respectively. Other values of γ between 0 and ∞ result in methods that lie on the continuum between OLS and PCR. The optimum value of γ and the number of basis functions in the model determine model generality, and are obtained via crossvalidation. The CR model is at least as accurate and compact as that obtained by OLS, PLS or PCR (Stone and Brooks, 1990; Wise and Ricker, 1993).

Unification of linear and nonlinear methods based on linear projection requires a common objective function that also includes nonlinear methods in the CR optimization criterion given by equation (7). This criterion cannot be applied directly to nonlinear empirical modeling methods based on linear projection, since the nonlinearity of the basis functions is not included in equation (7). The nonlinearity of basis functions does not change the optimization criterion used by PCR and NLPCR for determining the

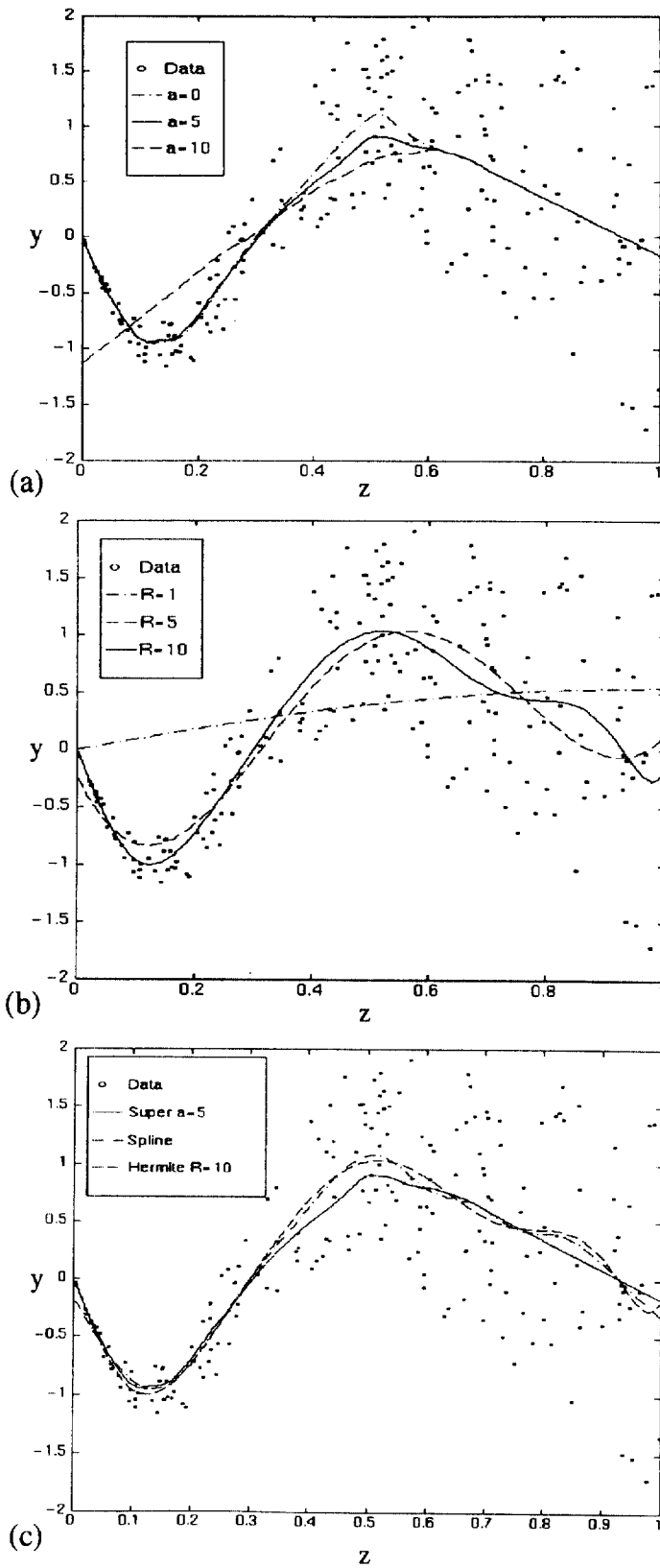


Fig. 2. Performance of techniques for determining adaptive basis functions. (a) Supersmoother for different values of tone control parameter. (b) Hermite polynomials of different orders. (c) Relative performance of each technique is quite similar.

projection directions, since both methods focus on transforming only the input space by maximizing the variance captured by the projected inputs as,

$$\max_{z_m} \{ \text{var}(\mathbf{X}\alpha_m) \}. \tag{8}$$

At the other extreme of the continuum of methods based on linear projection, are the techniques of OLS, PPR and BPN, since their optimization criterion focuses entirely on minimizing the output prediction error. This optimization criterion is equivalent to maximizing the square of the correlation between the actual and approximated outputs as stated by the following theorem:

Theorem. *The projection directions, α_m , of the m -th basis function that minimize the output mean-squares error of approximation,*

$$\max_{\alpha_m} \frac{1}{I} \sum_{i=1}^I (y_i - \hat{y}_i)^2 \tag{9}$$

are identical to the projection directions that maximize the square of the correlation between the output data points, and output of the m -th basis function,

$$\max_{z_m} \{ \text{corr}^2(\mathbf{y}, \theta_m(\mathbf{X}\alpha_m)) \}, \tag{10}$$

where, \hat{y}_i is given by equation (1).

This theorem is proved in the appendix by showing that equating the partial derivatives of equations (9) and (10) with respect to each projection direction results in identical equations for determining the optimum projection directions. Consequently, the optimization criterion for PPR and BPN may be written as equation (10). Furthermore, the optimization criteria given by equations (8) and (10) for PPR/BPN and NLPCR may be combined as

$$\max_{z_m} \{ \text{corr}^2(\mathbf{y}, \theta_m(\mathbf{X}\alpha_m)) \text{var}(\mathbf{X}\alpha_m) \} \tag{11}$$

and should result in a method that lies between PPR and NLPCR. Indeed, equation (11) has been used as the optimization criterion for NLPLS modeling by Wold *et al.* (1989) for quadratic PLS, Wold (1992) for spline PLS, and Holcomb and Morari (1992) for neural net/PLS. The NLPLS technique developed by Frank (1990) and Qin and McAvooy (1992) uses the linear PLS optimization criterion due to its computational ease for determining the optimum projection directions, and based on the assumption that the nonlinear activation functions do not have a significant effect on the optimum projection directions for NLPLS modeling.

Equations (8), (10) and (11) may be combined to obtain a general optimization criterion that subsumes all methods based on linear projection as

$$\max_{z_m} \{ [\text{corr}^2(\mathbf{y}, \theta_m(\mathbf{X}\alpha_m))] [\text{var}(\mathbf{X}\alpha_m)]^\gamma \}, \tag{12}$$

where values of γ equal to 0, 1, and ∞ result in BPN, PPR or OLS; NLPLS or PLS; and NLPCR or PCR, respectively. The exponents in equation (12) may be modified to,

$$\max_{z_m} \{ [\text{corr}^2(\mathbf{y}, \theta_m(\mathbf{X}\alpha_m))]^{1+\gamma-2\gamma^2} [\text{var}(\mathbf{X}\alpha_m)]^{-3\gamma-2\gamma^2} \}. \tag{13}$$

This objective function reduces to existing methods for $\gamma = 0, 0.5$, and 1, respectively, as summarized in Table 2. The remaining adjustable parameters in the empirical model, namely the regression coefficients, β_m and basis functions, θ_m are determined by minimizing the mean-squares error of approximation,

$$\min_{\beta_m, \theta_m} \frac{1}{I} \sum_{i=1}^I (y_i - \hat{y}_i)^2. \tag{14}$$

Equations (13) and (14) constitute the general objective function that unifies all methods based on linear projection.

The effect of the adjustable parameter, γ on the generality of the empirical model may be understood in terms of the bias-variance trade-off. The behavior of the bias and variance with changing values of γ is illustrated in Fig. 3. As γ increases from 0 to 1, the model bias increases, while the variance decreases, causing the mean-squares error of approximation to go through a minimum. The NLPCR training methodology aims to find this value of γ that optimizes the bias-variance trade-off as described in the next section.

5. Common hierarchical training methodology

The final challenge for the unification of empirical modeling methods based on linear projection is the development of a common training methodology that uses the general basis functions described in Section 3, and the common optimization criterion described in Section 4, to determine the empirical model in an efficient manner. Training methodologies for empirical model building belong to two main categories depending on how the model parameters are estimated.

- The *simultaneous* modeling approach determines all the model parameters together for the entire model. Examples of this approach include eigen value

Table 2. Specialization of NLPCR optimization criterion for projection directions to existing methods based on linear projection

γ	Linear basis functions	Nonlinear basis functions
0	OLS	PPR/BPN
1/2	PLS	NLPLS
1	PCR	NLPCR

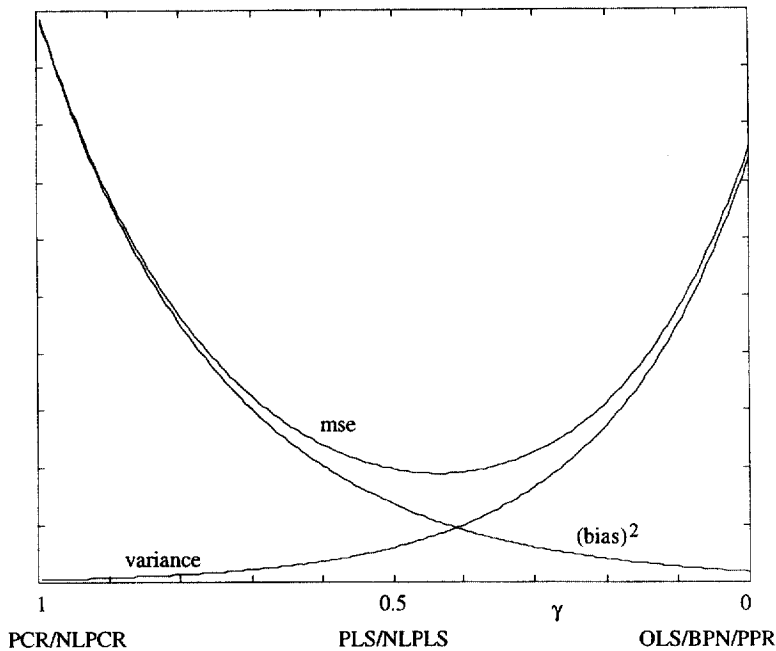


Fig. 3. Variation of model bias, variance, and mean-squares error with changing γ .

decomposition for computing the projection directions in PCR and PLS, and the error back-propagation algorithm for BPN (Rumelhart and McClelland, 1986).

- The *hierarchical* modeling approach determines model parameters for one basis function at a time, by approximating the residual of previously added basis functions. Examples of this approach include the nonlinear iterative partial least squares (NIPALS) algorithm (Martens and Naes, 1989) for PCR and PLS, cascade correlation for BPN (Fahlman and Lebiere, 1990), and the PPR algorithm (Friedman and Stuetzle, 1984).

Hierarchical modeling methods are usually more efficient than their simultaneous modeling counterparts since an existing model may be easily adapted by adding new nodes to capture the residual error of approximation as necessary. Such a hierarchical training method for NLPCR is developed in this section.

The steps comprising the hierarchical NLPCR training methodology, are shown in Table 3, depicted in Fig. 4, and described in the rest of this section. Models are developed for different values of the objective function parameter, γ , which is specified in Step (1). Before starting the modeling, the output residual, \mathbf{r}_1 is initialized as being equal to the output, \mathbf{y} , and the input residual, \mathbf{E}_1 is initialized as equal to the input, \mathbf{X} , as shown in Step (2). Initial values are assumed for the projection directions, α , and regression coefficients, β , for all the basis functions, and the basis function itself

is determined as the ratio of the output residual to corresponding regression coefficient. Steps (3)–(12) constitute the hierarchical node-by-node training method that approximates the output residual, \mathbf{r}_m , and if specified, the input residual, \mathbf{E}_m . The projection directions are computed in Step (5) by optimizing the general objective function for the selected value of γ , for the basis function and regression coefficient determined in the previous iteration. The optimum univariate basis function relating the latent variables, \mathbf{z}_m to the output residual, \mathbf{r}_m , is determined in Step (7) by the selected smoothing technique to minimize the output mean-squares error of approximation. The regression coefficient or output weight, β_m is computed in Step (8) as the product of the pseudo-inverse of \mathbf{z}_m and \mathbf{r}_m , to minimize the output residual mean-squares error. Steps (5)–(8) are repeated until convergence, to yield the parameters for the new node added to the model. The model prediction is updated in Step (10) by adding the contribution of the newly trained node. The output residual is updated in Step (11) by subtracting the prediction of the new node to obtain the residual to be approximated by the next node. If orthonormal projection directions are desired, as in PCR and PLS, then the input residual also needs to be updated as shown in Step (12a), otherwise, the input residual is left unchanged as shown in Step (12b). In general, more accurate models are obtained if the projection directions are not required to be orthonormal. If orthonormal projection directions or latent variables are not required and the model consists of more than one node, then the model parameters may

Table 3. Training methodology for nonlinear continuum regression

#	Description	
1	specify γ	For $\gamma = 0$ to 1,
2	Initialize	Initial guess for α, β $\mathbf{r}_1 = \mathbf{y}; \mathbf{E}_1 = \mathbf{X}, \theta_1 = \mathbf{r}_1/\beta_1$
3	Begin loop	For $m = 1$ to M
4	Start optimization	Until convergence,
5	Optimize projection	$\alpha_m = \max \{[\text{corr}^2(\mathbf{r}_m, \theta_m(\mathbf{E}_m \alpha_m))]\}^{1+\gamma-2\gamma^2} [\text{var}(\mathbf{E}_m \alpha_m)]^{3\gamma-2\gamma^2}$
	Direction	$\ \alpha_m\ = 1$
6	Latent variable	$\mathbf{z}_m = \mathbf{E}_m \alpha_m$
7	Activation function	$\theta_m(\mathbf{z}_m) = \frac{1}{\beta_m} \text{smooth}(\mathbf{z}_m, \mathbf{r}_m)$
8	Regression coefficient	$\beta_m = [\theta_m(\mathbf{z}_m)^T \theta_m(\mathbf{z}_m)]^{-1} \theta_m(\mathbf{z}_m) \mathbf{r}_m = \left[\frac{\sum_i \mathbf{r}_m \theta_m(\mathbf{z}_m)}{\sum_i \theta_m(\mathbf{z}_m)^2} \right]$
9	End optimization	end
10	Model update	$\hat{\mathbf{y}}_{m+1} = \hat{\mathbf{y}}_m + \beta_m \theta_m(\mathbf{z}_m)$
11	Output residual update	$\mathbf{r}_{m+1} = \mathbf{r}_m - \beta_m \theta_m(\mathbf{z}_m)$
12a	Input update	$\mathbf{E}_{m+1} = \mathbf{E}_m - \mathbf{z}_m \mathbf{z}_m^T$, or
12b		$\mathbf{E}_{m+1} = \mathbf{E}_m$
13	Backfitting	if $m > 1$, backfit previously added nodes
14	Termination	if $E[(\mathbf{y} - \hat{\mathbf{y}}_{m+1})^2] \leq \epsilon$, exit
15	End m loop	End
16	End γ loop	end

be adjusted to improve the contribution of each basis function in the model by backfitting or backward pruning.

Steps (1)–(12) in Table 3 introduce basis functions to minimize the residual output error without accounting for the nature of previously added basis functions. This “greedy” approach to empirical modeling and the nonorthogonal nature of the basis functions may not result in the best utilization of the approximation ability of all the basis functions together. Consequently, to obtain the most compact model with the best approximation ability, the parameters of previously added nodes should be adjusted while considering the contribution of the newly added nodes. Backfitting and backward pruning are two approaches for improving the contribution of each node.

- In *backfitting*, the parameters and basis functions of each previously added node are fine-tuned to capture the residual objective function not captured by any other node, while keeping all other nodes unchanged. This procedure is repeated for every new node after the first one, until the objective function captured by each node cannot be improved any further. The stopping criterion for adding new nodes in the model may be determined by crossvalidation with testing data. A commonly used heuristic is to stop adding new nodes when the prediction error for testing data increases for two consecutive nodes. The number of nodes that resulted in the smallest error for testing data are included in the

final model (Friedman, 1985; Roosen and Hastie, 1994). Unfortunately, due to the nonlinear nature of the model, the result of this forward growing approach may represent a local minimum.

- The alternate approach of modeling by *backward pruning* is claimed to be better at avoiding local minima (Friedman, 1985). This approach consists of first developing a model with more nodes than necessary without any backfitting. Then, the least significant nodes are eliminated, and the parameters of the remaining nodes are backfitted in the order of their importance. The significance of each node is determined by the magnitude of its regression coefficient, β_m . Based on empirical studies on PPR, Hwang *et al.* (1994) and Roosen and Hastie (1994) suggest developing a model with two more nodes than necessary followed by backward pruning.

Improved modeling by NLCR is due to the presence of the new adjustable parameter, γ . Consequently, efficient techniques for finding the best value of γ are essential for the application of NLCR modeling to practical problems. This value of γ may be found by determining several models for different values of γ between 0 and 1, and selecting the value of γ and number of basis functions that result in the smallest error of approximation for testing data. Unfortunately, the nonlinear nature of the model can make this approach computationally expensive as the number of training data, dimension of the input space, and number of basis functions increase. Furthermore,

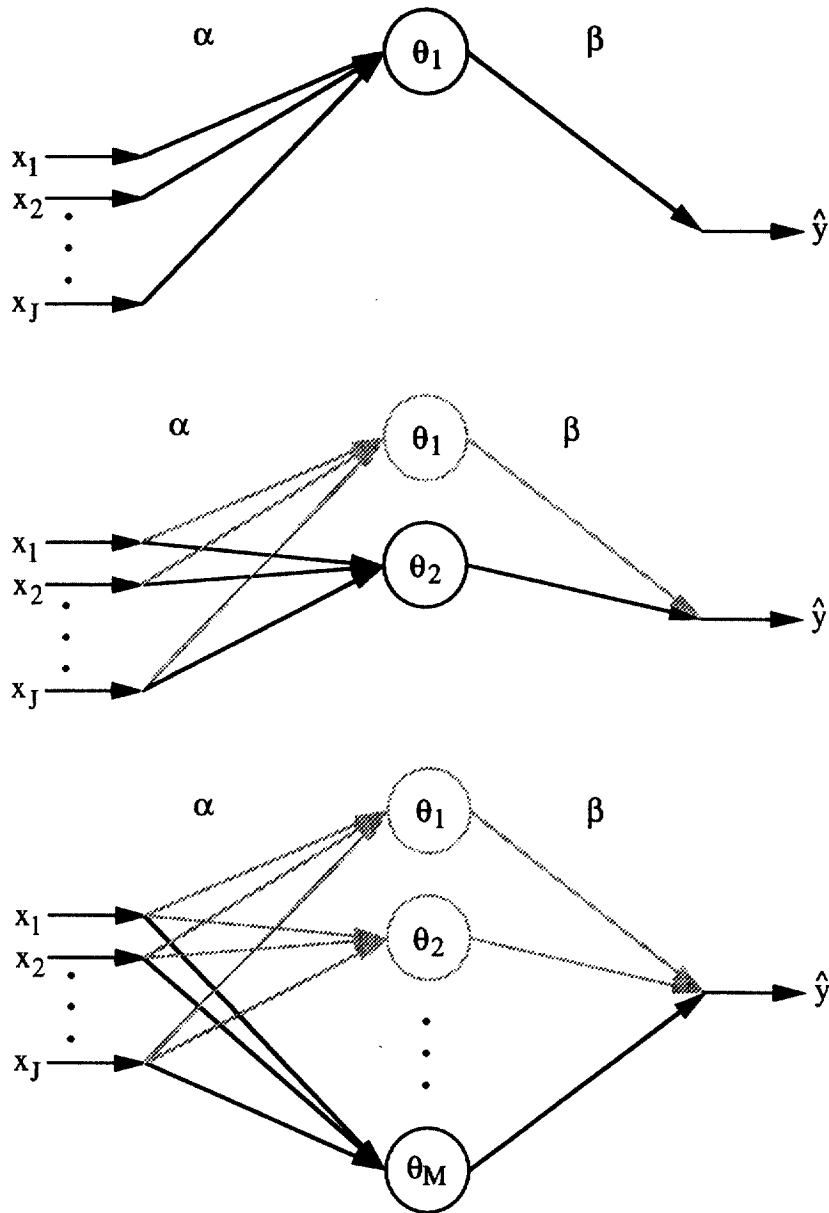


Fig. 4. Hierarchical training methodology for NLCR. Parameters corresponding to dark edges and nodes are optimized.

the nonlinear error surface may necessitate trial and error with several different initial values of the parameters to avoid local minima. These practical and computational issues may be addressed by exploiting the following properties of NLCR models.

- Unique values of the projection directions for $\gamma = 1$ may be determined by maximizing the variance captured by the projected inputs. If orthogonal projection directions are not required, then the projection directions for all nodes will be equal to the first principal component of the input data matrix.
- Decreasing the value of γ causes the projection directions to gradually rotate away from those cap-

turing the relationship between the inputs to those minimizing the output prediction error.

Thus, the NLCR model may be first determined for $\gamma = 1$, and the resulting parameters used as initial values of the parameters for modeling at smaller values of γ . This approach provides reproducible results and decreases the computation time for NLCR modeling.

The NLCR training methodology may be specialized to hierarchical algorithms for existing methods based on linear projection. For example, the NIPALS algorithm for PLS may be obtained by restricting the basis functions to be linear, selecting $\gamma = 0.5$, and

Table 4 Projection directions, regression coefficients, and mean-squares error for training and testing data for first node. Number of training data is 5, number of testing data is 95, $\sigma_3 = \sigma_4 = 1$

γ	α_{11}	α_{21}	α_{31}	α_{41}	β_1	MSE train	MSE test
1.0	0.6845	-0.6870	0.0866	-0.2280	0.8517	7.455e-02	6.147e-01
0.95	0.6891	-0.6937	0.0795	-0.1939	0.8562	6.701e-02	5.551e-01
0.9	0.6907	-0.6981	0.0826	-0.1696	0.8589	6.237e-02	5.364e-01
0.85	0.6909	-0.7014	0.0895	-0.1506	0.8609	5.887e-02	5.318e-01
0.8	0.6905	-0.7039	0.0979	-0.1348	0.8626	5.591e-02	5.362e-01
0.75	0.6897	-0.7058	0.1069	-0.1212	0.8641	5.333e-02	5.462e-01
0.7	0.6886	-0.7074	0.1163	-0.1092	0.8654	5.099e-02	5.618e-01
0.65	0.6873	-0.7085	0.1259	-0.0985	0.8667	4.883e-02	5.822e-01
0.6	0.6857	-0.7096	0.1360	-0.0882	0.8679	4.670e-02	6.078e-01
0.55	0.6839	-0.7104	0.1466	-0.0784	0.8691	4.463e-02	6.398e-01
0.5	0.6817	-0.7111	0.1580	-0.0689	0.8703	4.252e-02	6.806e-01
0.45	0.6793	-0.7113	0.1706	-0.0591	0.8716	4.032e-02	7.295e-01
0.4	0.6762	-0.7115	0.1848	-0.0489	0.8730	3.795e-02	7.916e-01
0.35	0.6725	-0.7112	0.2014	-0.0380	0.8745	3.531e-02	8.796e-01
0.3	0.6675	-0.7105	0.2215	-0.0259	0.8762	3.228e-02	1.000e + 00
0.25	0.6607	-0.7088	0.2469	-0.0121	0.8783	2.866e-02	1.196e + 00
0.2	0.6508	-0.7055	0.2804	0.0043	0.8808	2.417e-02	1.585e + 00
0.15	0.6355	-0.6990	0.3272	0.0243	0.8841	1.842e-02	2.510e + 00
0.1	0.6100	-0.6858	0.3941	0.0488	0.8882	1.114e-02	5.701e + 00
0.05	0.5714	-0.6636	0.4771	0.0740	0.8922	4.073e-03	2.526e + 01
0.0	0.5146	-0.6276	0.5759	0.0982	0.8940	7.871e-04	2.134e + 07

determining the input and output residuals after training each node. Backfitting is not needed since the projection directions are fixed by the orthogonality requirement. Several existing versions of nonlinear PLS such as, quadratic PLS (Wold *et al.*, 1989), supersmoothing PLS (Frank, 1990), or neural network PLS (Qin and McAvoy, 1992), may be obtained by using the appropriate PLS objective function, and computing the activation functions using a quadratic model, supersmoothing, or backpropagation networks, respectively. Specializing the general method to PPR, requires determining the projection directions, basis functions, and regression coefficients by maximizing the objective function for $\gamma = 0$, and computing the output residual only. Restricting the basis functions for PPR to sigmoids produces a hierarchical algorithm for BPN that is similar to the cascade correlation approach (Fahlman and Lebiere, 1990).

6. Illustrative examples

The properties of NLCR are illustrated by the following examples based on synthetic and industrial data. The first example models an inherently three-dimensional surface to permit easy visualization of the model, and gain insight into the effect of the NLCR parameter, γ . The second example demonstrates the ability of NLCR to model industrial data. The NLCR training methodology is implemented in Matlab, and is available from the corresponding author.

Parabola example

This set of illustrative examples models a parabola, with the variables contaminated by random error,

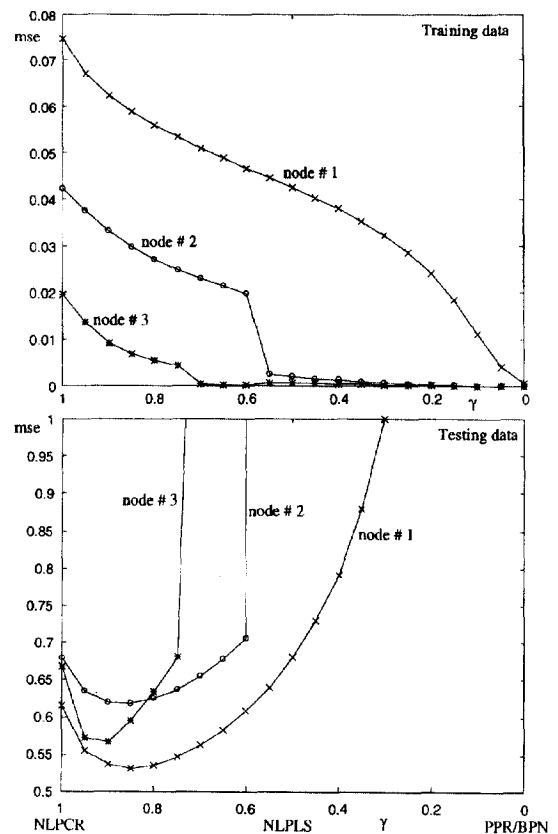


Fig. 5. Mean-squares error for NLCR modeling of parabola example with 5 training data, $\sigma_3 = \sigma_4 = 1$.

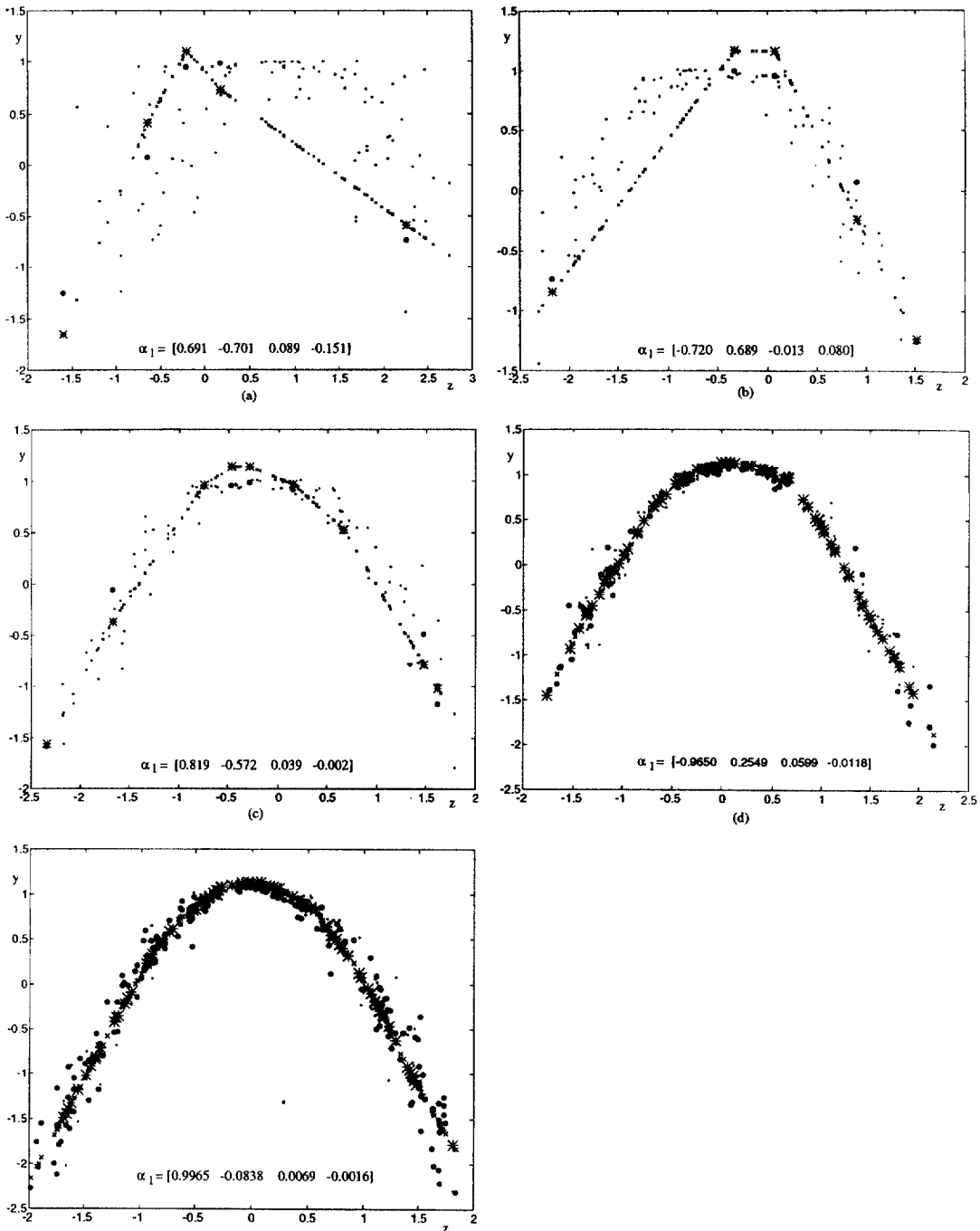


Fig. 6. Basis functions for parabola example. (a) 5 training data, $\sigma_3 = \sigma_4 = 1$, $\gamma = 0.85$; (b) 5 training data, $\sigma_3 = \sigma_4 = 0.1$, $\gamma = .15$; (c) 10 training data, $\sigma_3 = \sigma_4 = 0.1$, $\gamma = 0.1$; (d) 50 training data, $\sigma_3 = \sigma_4 = 0.1$, $\gamma = 0.02$; (e) 189 training data, $\sigma_3 = \sigma_4 = 0.1$, $\gamma = 0$. Legend — (●) training data; (•) testing data; (■) basis function (training); (*) basis function (testing).

and two variables being pure noise,

$$y = t_1^2, \quad x_1 = t_1 + 0.1\epsilon_1, \quad x_2 = t_2 + 0.1\epsilon_2,$$

$$x_3 = \sigma_3\epsilon_3, \quad x_4 = \sigma_4\epsilon_4,$$

where, $\epsilon_i, i = 1, \dots, 4$, denote independent and identical Gaussian white noise with unit variance, σ_3 and

σ_4 denote the standard deviation of variables x_3 and x_4 , respectively, and t_1 and t_2 are approximately linearly related. This model indicates that the underlying relationship between the two relevant inputs and the output is a three-dimensional parabola, with an optimum projection direction of $[1 \ 0 \ 0 \ 0]$ for noise-free variables. The performance of NLCR is compared for

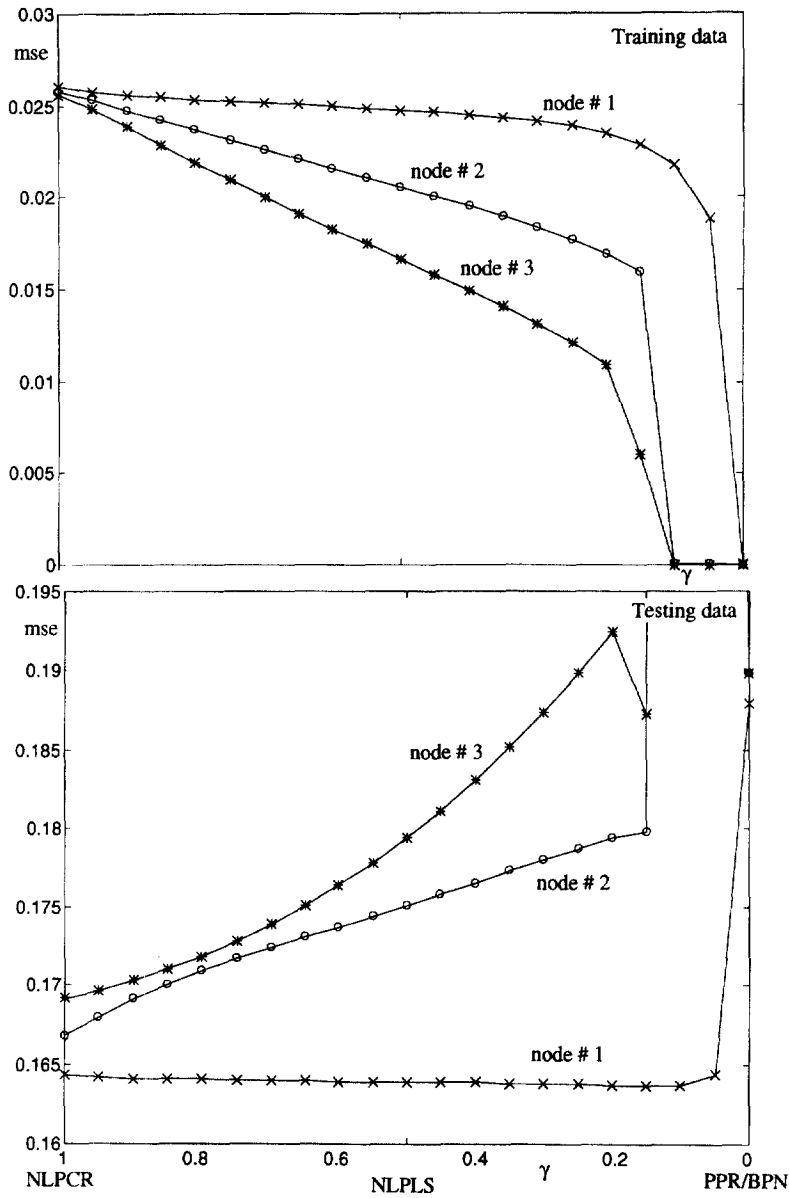


Fig. 7. Mean-squares error for NLPCR modeling of parabola example with 5 training data, $\sigma_3 = \sigma_4 = 0.1$.

different amounts of training data, and different variance of the irrelevant variables. For each example, the same set of testing data are used, consisting of 95 data points. Such a large amount of testing data are selected to evaluate the ability of each example to capture the underlying hypersurface. All the inputs are scaled to have a zero mean, with x_1 and x_2 of unit standard deviation, and x_3 , and x_4 of standard deviation equal to the selected values of σ_3 and σ_4 , respectively. For each case study described below, the models were determined by trial-and-error with random initialization of the model parameters, and by using the results at adjacent values of γ for initialization. Both approaches yielded similar results. The models in each

case study were developed with seven basis functions. Since the best model required no more than three basis functions, the results reported below are for a maximum of three basis functions.

Case Study 1: Five training data, $\sigma_3 = \sigma_4 = 1$. The projection directions, regression coefficient, and errors on training and testing data for the first node are shown in Table 4. As expected, the projection directions change the orientation of the projection hyperplane from that maximizing the captured variance for $\gamma = 1$, to that minimizing the prediction error for the training data at $\gamma = 0$. The mean-squares error of approximation for training and testing data for three basis functions are plotted in Fig. 5. The error of

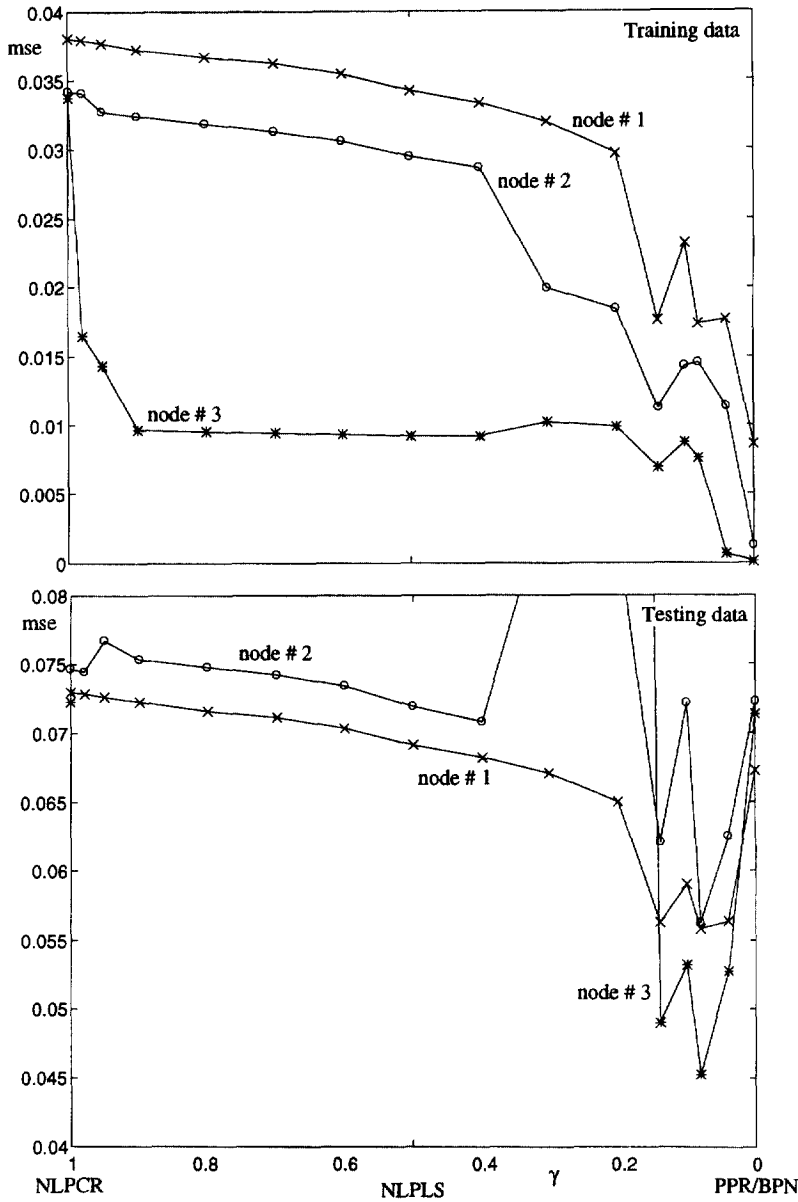


Fig. 8. Mean-squares error for NLCR modeling of parabola example with 10 training data, $\sigma_3 = \sigma_4 = 0.1$.

approximation for training data decreases with decreasing values of γ and increasing number of nodes, but the error of approximation on testing data goes through a minimum at $\gamma = 0.85$ with one node. This optimum NLCR model is significantly better than that obtained by the existing methods of PPR/BPN at $\gamma = 0$, NLPLS at $\gamma = 0.5$, and NLPCR at $\gamma = 1$. As shown in Fig. 5 and the last column of Table 4, the performance of PPR is several orders of magnitude worse than that of other methods based on linear projection. Table 4 also shows that the projection directions for $\gamma = 0.85$ are closest to the ideal directions of [1000]. The available training and testing

data and the basis function in the first node at $\gamma = 0.85$ are depicted in Fig. 6a.

Case Study 2: Five training data, $\sigma_3 = \sigma_4 = 0.1$. Results for NLCR modeling with the same five training data as in Case Study 1, but after decreasing the value of σ_3 and σ_4 to 0.1 are shown in Fig. 7. The behavior of the training and testing errors is similar to that in Fig. 5, but the smallest error of approximation for testing data is obtained for $\gamma = 0.15$ with one basis function. The smaller optimum value of γ indicates that due to the diminished contribution of the irrelevant variables, capturing the relationship between the inputs is less important than in Case Study 1. The

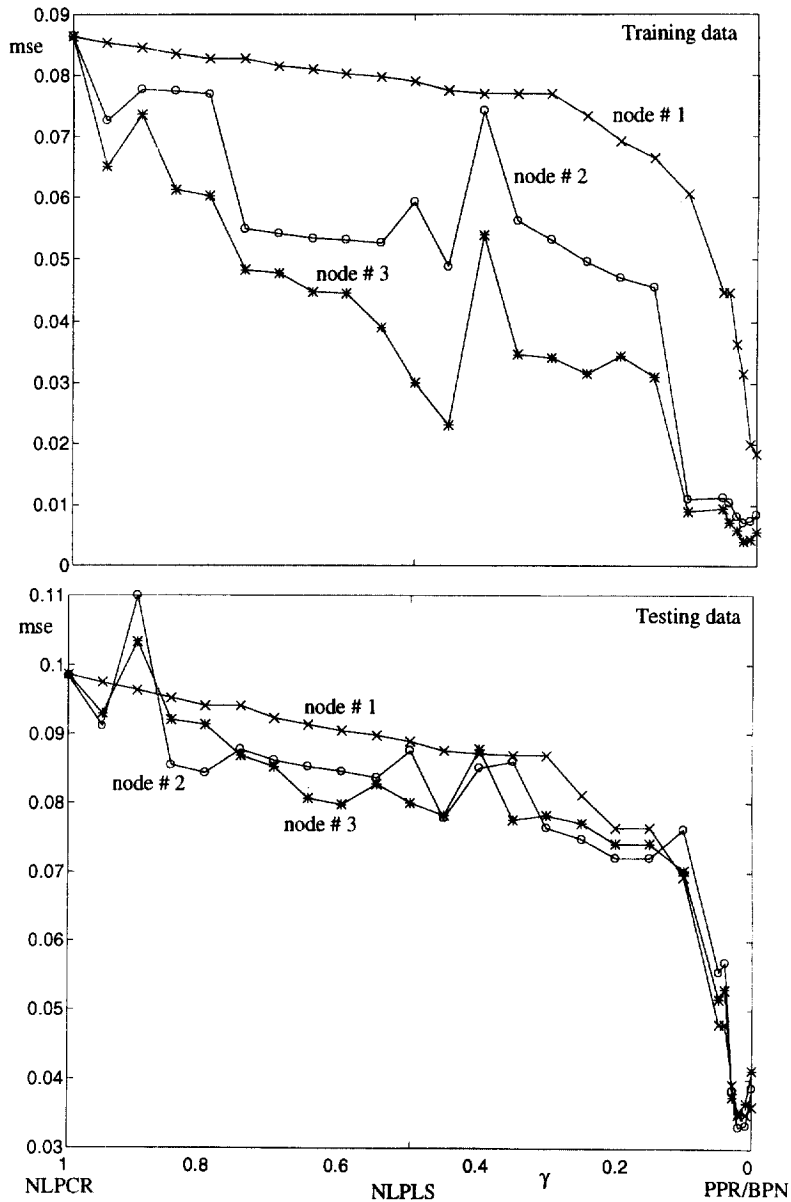


Fig. 9. Mean-squares error for NLPCR modeling of parabola example with 50 training data, $\sigma_3 = \sigma_4 = 0.1$.

training data, testing data, and basis function values in the projected input-output space are depicted in Fig. 6b. The optimum projection direction for this case study is closer to $[1000]$ than that for Case Study 1, and the basis function in Fig. 6b is closer to a parabola than that in Fig. 6a.

Case Study 3: Ten training data, $\sigma_3 = \sigma_4 = 0.1$. The errors with 10 data points used for training are shown in Fig. 8. The smallest error of approximation for testing data is obtained for $\gamma = 0.1$ with three nodes. The basis function and projection directions for the optimum model are shown in Fig. 6c. As the number of training data increase, the optimum value of γ tends towards 0 indicating a decreasing need for a biased

model, the optimum projection directions are closer to $[1000]$ than for Case Studies 1 and 2, and the basis function starts looking more like the underlying parabola.

Case Study 4: Fifty training data, $\sigma_3 = \sigma_4 = 0.1$. The results for modeling with 50 training data are shown in Figs 9 and 6d. The best model is obtained for $\gamma = 0.02$ with two nodes, which is only slightly better than the model at $\gamma = 0$. The result of this case study continues the trend towards smaller optimum γ with increasing amounts of training data. The optimum projection is also much closer to $[1000]$ than the previous case studies, and the basis function looks much more like a parabola, as depicted in Fig. 6d.

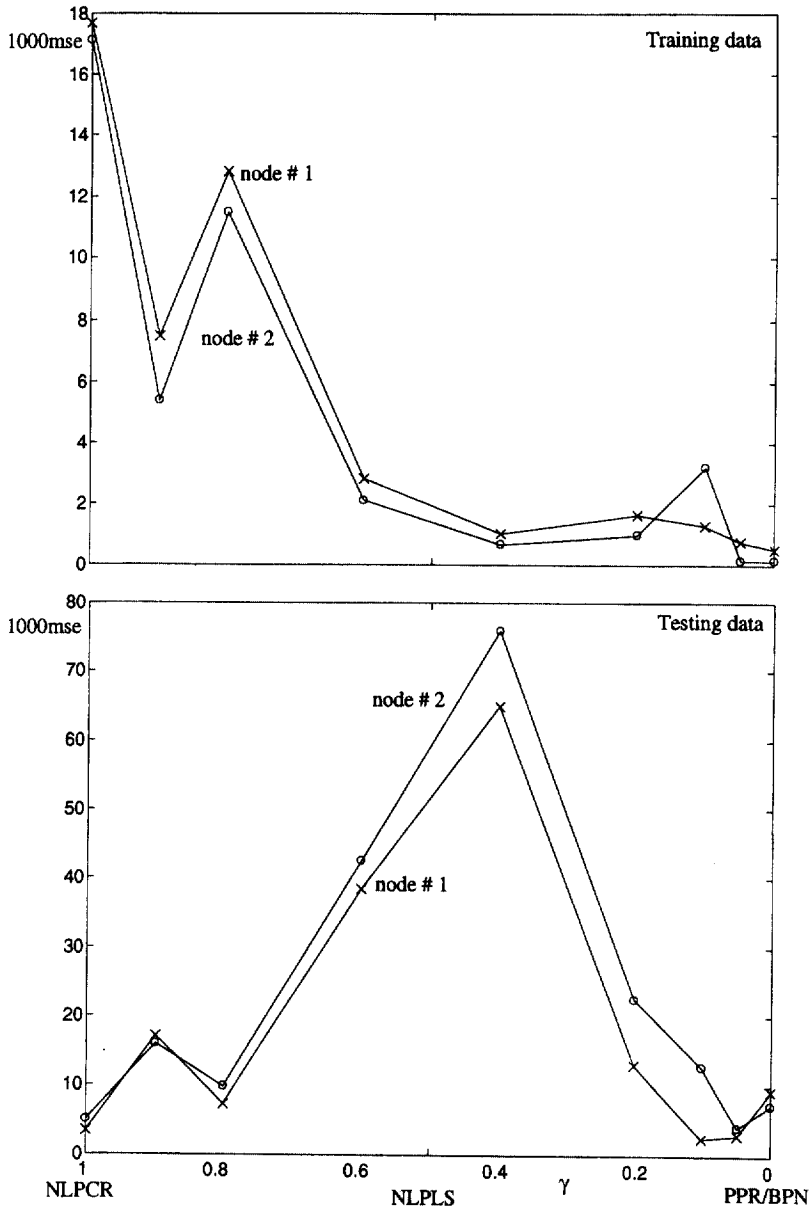


Fig. 10. Mean-squares error for polymerization example. Best model is for $\gamma = 0.1$.

Increasing the number of training data to 189 results in the best model for $\gamma = 0$, with the basis function and projection direction closest to the optimum, as depicted in Fig. 6e.

Industrial polymerization example

This example involves modeling of the relationship between ten input (predictor) variables and four output (response) variables based on data collected from the pilot plant of a polymerization process. The available data consist of 61 observations. Due to the proprietary nature of this example, it is not possible to provide any more information about the process. In this paper, an empirical model is found between the

inputs and the first output variable only. This example has also been used by DeVeaux *et al.* (1993) to compare modeling by MARS with that by BPN, but their results cannot be compared directly with this case study since this case study is based on using 51 observations for training and 10 for testing, whereas DeVeaux *et al.* used a jackknifing procedure for determining the most general model.

The results of NLPCR modeling at different values of γ are shown in Fig. 10. The smallest error of approximation on testing data is obtained for $\gamma = 0.1$ with one node. This error is almost an order of magnitude smaller than that obtained by PPR/BPN at $\gamma = 0$, and about half of that obtained by NLPCR at $\gamma = 1$, thus

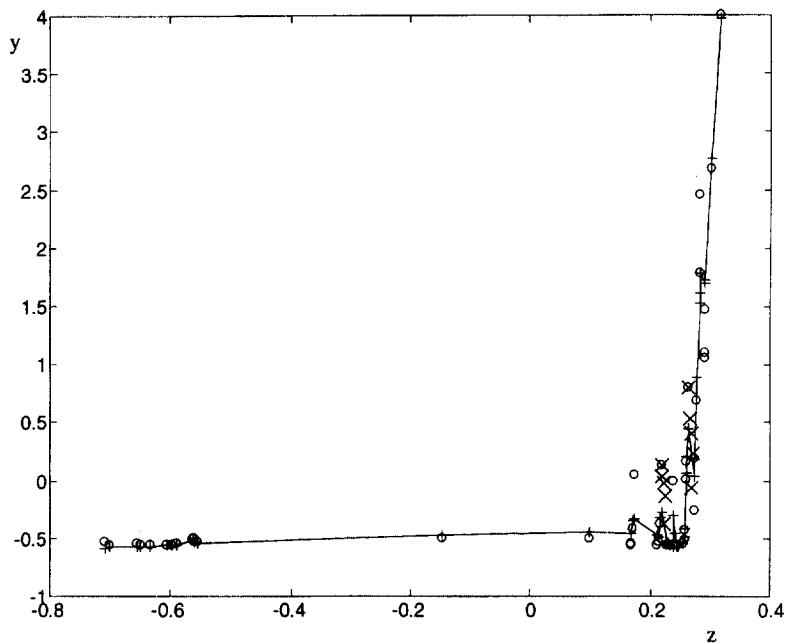


Fig. 11. Projected input-output space for most general NLCR model of polymerization example at $\gamma = 0.1$. (○) Training data; (×) testing data; (+) basis function.

illustrating the improved modeling by NLCR as compared to modeling by existing methods based on linear projection. The training and testing data and basis function for the best model for $\gamma = 0.1$ are shown in Fig. 11. Several samples in this projected input-output space are quite far from the main cluster of data, indicating that they may be outlying observations, as discussed by DeVeaux *et al.* (1993). The NLCR method may be made less sensitive to outlying observations by replacing its objective functions by their robust counterparts.

7. Conclusions and discussion

This paper describes the unification of empirical modeling methods that combine the inputs by linear projection before application of the basis functions. The unification is based on the insight provided by the common framework for neural and statistical empirical modeling methods (Bakshi and Utojo, 1998). The result is a new empirical modeling method called nonlinear continuum regression (NLCR) that subsumes existing methods based on linear projection including, OLS, PLS, PCR, BPN, PPR, NLPLS, and NLPCR. The NLCR model utilizes a common technique for determining basis functions of any shape in the projected input-output space, a general objective function for all methods based on linear projection, and an efficient hierarchical training methodology. The basis functions are determined by univariate smoothing methods such as, variable span smoothers, splines and polynomials. The common objective func-

tion introduces a new adjustable parameter, γ , to span the continuum of methods from NLPCR to PPR. Existing methods based on linear projection may be obtained by setting γ to 0 for PPR, BPN and OLS, to 0.5 for NLPLS and PLS, and to 1 for NLPCR and PCR. The NLCR parameter, γ , complements the effect of the number of basis functions on the bias of the empirical model. This additional degree of freedom for optimizing the bias-variance trade-off results in models that are more general and more compact than those obtained by existing methods based on linear projection. Thus, NLCR is able to automatically select the best method based on linear projection for a given task without requiring arbitrary or subjective decisions by the user.

An efficient hierarchical training method is developed for NLCR that trains one node at a time. The efficiency and convergence to local minima of NLCR modeling may be addressed by initializing the model parameters based on results at adjacent values of γ . The NLCR methodology is illustrated by empirical modeling of data from synthetic and industrial examples. These examples indicate that the NLCR technique can be applied to any empirical modeling problem, to obtain better models than those obtained by existing methods based on linear projection. The benefits of NLCR models are likely to be most significant for problems where the variables are related to each other, and limited quantities of training data are available. Application of NLCR to a variety of empirical modeling problems of practical and theoretical interest, and more efficient methods for selecting the

best value of γ are currently being explored. Since empirical models are essential for several engineering tasks, this unified modeling method is likely to be useful beyond the area of chemical process operation and control.

Empirical modeling by methods based on linear projection is known to suffer from several disadvantages such as, extrapolation outside the region of available data without warning, and computationally expensive adaptation to new data. These shortcomings are due to the nonlocal nature of the projection hyperplane and may be overcome by local methods based on nonlinear projection that project the data on a localized hypersurface such as, clustering methods, radial basis function networks, and wavelet networks. The approach for unifying empirical modeling methods based on linear projection presented in this paper and the insight provided by the common framework for empirical modeling methods may be used for unifying local methods based on nonlinear projection by developing the appropriate common objective function and training methodology. Development of such a unified technique is expected to provide better models than existing methods for classification problems such as fault diagnosis, and further reduce the arbitrariness in selecting the appropriate method for a given task.

Acknowledgements

The authors acknowledge the partial financial support from an Ohio State University Seed Grant and the Ohio Aerospace Institute through grant number CCRP-95-1-029, Profs R.D. DeVaux and L. H. Ungar for providing the data for the polymerization example, and Mr. Zafar Ali for help in completing the case studies.

Nomenclature

E	input residual matrix
I	number of measurements
J	number of inputs
K	number of outputs
M	number of nodes
Q	order of Hermite polynomial
r_m	output residual vector approximated by m th node
\hat{x}	approximated input
x_{ij}	element of \mathbf{X}
x_j	j th column of \mathbf{X}
x_i^T	i th row of \mathbf{X}
\mathbf{X}	input or predictor variables matrix, $I \times J$
\hat{y}	approximated output
\mathbf{Y}	output or response variables matrix, $I \times K$
z_m	m th latent variable vector
\mathbf{Z}	latent variables matrix, $I \times M$
α_{jm}	input edge weight or projection directions connecting j th input to m th node
α	projection directions matrix, $J \times M$

β_{mk}	output edge weight connecting m th node to k th output
γ	NLCR objective function parameter
ϕ_m	input transformation function in m th node
θ_m	m th node or basis function
$\ \cdot\ $	Sum of the squares of each element in the argument vector

References

- Bakshi, B.R. and Utojo, U. (1998). A common framework for the unification of neural and statistical modeling methods, *Anal. Chim. Acta.*, submitted.
- Barron, A.R. and Barron, R.L. (1988). Statistical learning networks: a unifying view, In: Wegman, E.J., Gantz, D.T. and Miller, J.J. (Eds), *Computing Science and Statistics*.
- deJong, S. and Farebrother, R.W. (1994). Extending the relationship between ridge regression and continuum regression. *Chemom. Intell. Lab. Sys.*, **25**, 179–181.
- DeVaux, R.D., Psichtogis, D.C. and Ungar, L.H. (1993). A comparison of two nonparametric estimation schemes: MARS and neural networks, *Comput. Chem. Engng* **17**(8), 819.
- Fahlman, S.E. and Lebiere, C. (1990). The cascaded-correlation learning architecture. *Advances in Neural Information Processing Systems*, Vol. 2, pp. 524–532. Morgan Kaufmann, Los Altos, CA.
- Frank, I.E. (1990). A nonlinear PLS model. *Chemom. Intell. Lab. Systems*, **8**, 109–119.
- Frank, I.E. (1995). Modern nonlinear regression methods. *Chemom. Intell. Lab. System*, **27**, 1–9.
- Friedman, J.H. and Stuetzle, W. (1981). Projection pursuit regression. *J. Amer. Statist. Assoc.* **76**(376), 817–823.
- Friedman, J.H. (1984). A variable span smoother, Technical Report No. 5, Dept. of Statistics, Stanford University, Stanford, CA.
- Friedman, J.H. (1985). Classification and multiple regression through projection pursuit, Technical Report No. 12, Dept. of Statistics, Stanford University, Stanford, CA.
- Haario, H. and Taavitsainen, V.-M. (1994). Nonlinear data analysis. II. Examples of new link functions and optimization aspects, *Chemom. Intell. Lab. Systems* **23**, 51–64.
- Haykin, S.S. (1994). *Neural Networks: A Comprehensive Foundation*. Macmillan, New York.
- Holcomb, T.R. and Morari, M. (1992). PLS/neural networks. *Comput. Chem. Engng* **16**(4), 393–411.
- Hwang, J.-N., Lay, S.-R., Maechler, M., Martin, R.D. and Schimert, J. (1994). Regression modeling in back-propagation and projection pursuit learning. *IEEE Trans. Neural. Networks*. **5**(3), 342.
- Kosko, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, Englewood Cliffs, NJ.
- Kresta, J., MacGregor, J.F. and Marlin, T.E. (1991). Multivariate statistical monitoring of process operating performance, *Can. J. Chem. Engng* **69**, 35–47.
- Lorber, A., Wangen, L.E. and Kowalski, B.R. (1987). A theoretical foundation for the PLS algorithm, *J. Chemometrics* **1**, 19–31.
- Martin, E.B., Morris, A.J. and Zhang, J. (1995). Artificial neural networks and multivariate statistics. In: (Ed.) Bulsari, A.B. (Ed.) *Neural Networks for Chemical Engineers*.

- Martens, H. and Naes, T. (1989). *Multivariate Calibration*. Wiley, New York.
- Piovoso, M.J. and Owens, A.J. (1986). Sensor data analysis using artificial neural networks. In: (Eds.) Arkun, Y. and Ray, W.H., *Chemical Process Control CPC IV*. CACHE, Austin, TX.
- Qin, S.J. and McAvoy, T.J. (1992). Nonlinear PLS Modeling Using Neural Networks, *Comput. Chem. Engng.* **16**(4), 379–391.
- Roosen, C.B. and Hastie, T.J. (1994). Automatic smoothing spline projection pursuit. *J. Comput. Graph. Statist.* **3**(3), 235–248.
- Rumelhart, D.E., McClelland, J.L. et al. (1986). *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA.
- Sjoberg, J., Zhang, Q., Ljung, L., Benveniste, A., Deylon, B., Glorennec, P., Hjalmarsson, H. and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica* **31**(12), 1691–1724.
- Stone, M. and Brooks, R.J. (1990). Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *J. Roy. Statist. Soc. B* **52**(2), 237–269.
- Sundberg, R. (1993). Continuum regression and ridge regression. *J. Roy. Statist. Soc. B* **55**(3), 653–659.
- Utojo, U. (1996). A unified view of neural networks and multivariate statistical methods for empirical modeling. M.S. Thesis, The Ohio State University, Columbus, OH.
- Venkatasubramanian, V., Vaidyanathan, R. and Yamamoto, Y. (1990). Process fault detection and diagnosis using neural networks—I. Steady-state processes. *Comput. Chem. Engng* **14**(7), 699–712.
- Wise, B.M. and Ricker, N.L. (1993). Identification of finite impulse response models with continuum regression. *J. Chemom.* **7**, 1–14.
- Wold, S. (1982). Soft modeling. The basic design and some extensions. In: (Eds) Joreskog, K. and Wold, H. *Systems Under Indirect Observation*. Elsevier, Amsterdam.
- Wold, S. (1992). Nonlinear partial least squares modeling II. Spline inner relation. *Chemom. Intell. Lab. Systems* **14**, 71–84.
- Wold, S., Kettaneh-Wold, N. and Skagerberg, B. (1989). Nonlinear PLS modeling. *Chemom. Intel. Lab. Systems* **7**, 53–65.

Appendix

Theorem. The projection directions, α_m , obtained by minimizing the mean-squares error of approximation,

$$\min_{\alpha_m} \frac{1}{I} \sum_{i=1}^I (y_i - \hat{y}_i)^2$$

are identical to those obtained by maximizing the square of the correlation between the model output and basis function output,

$$\max_{\alpha_m} \{\text{corr}^2(\mathbf{y}, \theta_m(\mathbf{X}\alpha_m))\},$$

where, \hat{y} is given by equation (1).

Proof: Consider each basis function as fitting the residual output error of approximation, \mathbf{y} . Then, from

equation (1), the mean-squares error is

$$e = \frac{1}{I} \sum_{i=1}^I \left[y_i - \beta_m \theta_m \left(\sum_{j=1}^J \alpha_{jm} x_{ij} \right) \right]^2. \quad (\text{A1})$$

Represent the basis function output for the i th set of measurements as,

$$C_i = \theta_m \left(\sum_{j=1}^J \alpha_{jm} x_{ij} \right). \quad (\text{A2})$$

Taking the partial derivative of the mean-squares error given by equation (A1) with respect to α_{jm} ,

$$\begin{aligned} \frac{\partial e}{\partial \alpha_{jm}} &= \frac{\partial}{\partial \alpha_{jm}} \frac{1}{I} \sum_{i=1}^I (y_i - \beta_m C_i)^2 \\ &= 2 \sum_i (y_i - \beta_m C_i) \left(-\beta_m \frac{\partial C_i}{\partial \alpha_{jm}} \right) \\ &= -2\beta_m \left[\sum_i y_i \frac{\partial C_i}{\partial \alpha_{jm}} - \beta_m \sum_i C_i \frac{\partial C_i}{\partial \alpha_{jm}} \right]. \quad (\text{A3}) \end{aligned}$$

Equating the partial derivative of the mean-squares error to zero gives

$$\left[\sum_i y_i \frac{\partial C_i}{\partial \alpha_{jm}} - \beta_m \sum_i C_i \frac{\partial C_i}{\partial \alpha_{jm}} \right] = 0. \quad (\text{A4})$$

The squared of the correlation may be written as

$$r = \frac{\left(\sum_i y_i C_i \right)^2}{\left(\sum_i y_i^2 \right) \left(\sum_i C_i^2 \right)}. \quad (\text{A5})$$

Taking the partial derivative of equation (A5) with respect to α_{jm} gives

$$\begin{aligned} \frac{\partial r}{\partial \alpha_{jm}} &= \frac{\sum_i y_i C_i}{\sum_i y_i^2 \sum_i C_i^2} \left[2 \sum_i y_i \frac{\partial C_i}{\partial \alpha_{jm}} \right. \\ &\quad \left. - \frac{2 \sum_i y_i C_i}{\sum_i C_i^2} \sum_i C_i \frac{\partial C_i}{\partial \alpha_{jm}} \right]. \quad (\text{A6}) \end{aligned}$$

Substituting

$$\beta_m = (C^T C)^{-1} C^T y = \frac{\sum_i y_i C_i}{\sum_i C_i^2}$$

in equation (A6) and equating the derivative to zero yields,

$$\left[\sum_i y_i \frac{\partial C_i}{\partial \alpha_{jm}} - \beta_m \sum_i C_i \frac{\partial C_i}{\partial \alpha_{jm}} \right] = 0$$

which is identical to equation (A4), indicating that at their extrema, equations (9) and (10) will yield the same projection directions.