



ELSEVIER

Analytica Chimica Acta 384 (1999) 227–247

ANALYTICA
CHIMICA
ACTA

A common framework for the unification of neural, chemometric and statistical modeling methods

Bhavik R. Bakshi*, Utomo Utojo

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

Received 7 April 1998; received in revised form 2 October 1998; accepted 19 October 1998

Abstract

Extraction of empirical models from measured data is essential for several chemometric and engineering tasks. Selection of an appropriate method for a given task requires deep understanding of the characteristics of a variety of empirical modeling methods that have been derived from diverse fields such as statistics, chemometrics, and artificial intelligence. Unfortunately, the necessary insight into the plethora of empirical modeling methods is not easily available, making the selection process subjective and heuristic, often resulting in inferior empirical models. Furthermore, the properties of various methods are complementary, and combining these methods can result in better models. This paper presents a common framework for understanding the similarities and differences between various empirical modeling methods, and for developing hybrid techniques that combine the best properties of existing methods. The framework is based on representing all empirical modeling methods as a weighted sum of basis functions, and comparing various methods depending on decisions about the nature of the input transformation, type of activation functions, and optimization criteria. All empirical modeling methods transform the inputs by projection on a linear or nonlinear hypersurface or by selecting a subset of variables. The activation functions are of fixed or adaptive shape, and the optimization criteria for determining the model parameters are based on either the input space only, or both input and output space. An overview of several popular methods and an illustrative example are presented to enhance the insight into empirical modeling methods provided by the common framework. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Empirical modeling; Neural networks; Linear regression; Nonlinear regression

1. Introduction

The wide variety of methods popular for empirical modeling have been derived from several fields including statistics, chemometrics, and artificial intelligence. These methods may be broadly classified into three categories.

- *Linear statistical methods.* Linear multivariate statistical methods such as ordinary least squares (OLS), principal component analysis (PCA), principal component regression (PCR), partial least squares (PLS), and ridge regression (RR) are extremely popular and successful. These techniques are used for multivariate calibration in spectroscopy [1], process monitoring [2], fault detection and diagnosis [3], and dynamic model identification [4]. The linear model is physically interpretable,

*Corresponding author. Tel.: +1-614-292-4904; fax: +1-614-292-3769; e-mail: bakshi.2@osu.edu

and may provide useful insight into the system being modeled such as the relative importance of each variable in predicting the output. Estimation of the model parameters is efficient, and may be performed hierarchically for one node or basis function at a time, or simultaneously for all nodes. Most applications of linear statistical methods are for static, non-adaptive modeling, although modifications for recursive and adaptive modeling have been devised [5,6]. Techniques for estimating error bounds, robustness to outliers, compensation of missing data, and reliable modeling have also been developed [7,8]. Despite their advantages, application of linear multivariate statistical techniques is limited by their inability to model nonlinear relationships.

- *Artificial neural networks.* Artificial neural network modeling is inspired by artificial intelligence research, and has been a popular technique for nonlinear empirical modeling of chemical systems. Backpropagation network (BPN) and radial basis function networks (RBFNs) are most popular, but other networks such as adaptive resonance theory (ART), ellipsoidal basis function networks (EBFNs) and wavelet networks (wave-nets), are also being used. Neural networks have found wide application for determining the correlation between different types of spectra and chemical structure [9], investigation of the quantitative structure activity relationship [10], fault detection and diagnosis [11], nonlinear system identification [12], process control [13], and several applications in analytical chemistry [14]. The appeal of ANN lies in their universal approximation ability, parallel processing and recurrent dynamic modeling, but the model learned by ANN is usually black box in character, often requires a large ratio of training data to input variables, and learning is computationally expensive due to network construction based on simultaneous computation of all the model parameters.
- *Nonlinear multivariate statistical methods.* Many properties of linear statistical methods have been extended to nonlinear modeling by nonlinear multivariate statistical methods such as nonlinear principal component analysis (NLPCA), nonlinear principal component regression (NLPCR), nonlinear partial least squares regression (NLPLS), projection pursuit regression (PPR), classification

and regression trees (CART), and multivariate adaptive regression splines (MARS). Application of nonlinear statistical methods is relatively limited and recent, and includes multivariate calibration [15], quantitative structure–activity relationship modeling [16], process monitoring and supervision [17,18], and nonlinear system identification [19]. Like ANN, nonlinear statistical methods are also universal approximators, and like linear statistical methods, the model is often physically interpretable. These methods often perform well with a relatively small amount of training data, but their training methodology is best suited to batch, or off-line learning, and they have not received as much attention as linear methods or artificial neural networks for empirical modeling.

Given this plethora of methods, the user is faced with the vexing task of selecting the best method for a given empirical modeling problem. Selection of the best technique requires the user to have a deep understanding of all the modeling techniques with their advantages and disadvantages, and significant insight into the nature of the measured data and the process being modeled. Unfortunately, neither the deep insight into all the neural and statistical modeling techniques, nor the information about the nature of the modeling task are easily accessible. Consequently, selection of the appropriate empirical modeling method has become an art, involving the use of ad hoc selection criteria based on the user's bias or on "folklore". A common framework that brings out the similarities and differences between various empirical modeling methods can be invaluable in improving their understanding and in selecting the appropriate method for a given task.

A close look at neural and statistical modeling methods reveals that several of their properties are complementary in nature. For example, linear statistical methods are usually more physically interpretable than ANN, and are based on a firm theoretical foundation which facilitates robust modeling, estimation of error bounds, and compensation of missing data. In contrast, neural networks are well-suited for largescale and parallel computation, recursive modeling, and continuous adaptation, but are black box in nature. Artificial neural networks such as BPN often require a large amount of training data to obtain an

acceptable model for a given number of input variables, while statistical methods such as NLPCR and NLPLS can perform equally well with a smaller ratio of training data to input variables. These complementary properties of neural and statistical methods point towards the potential benefits of combining various empirical modeling methods. Systematic development of hybrid empirical modeling methods may be facilitated by a common framework for comparing empirical modeling methods.

Over the last several years, insight into the properties of and relationships between various empirical modeling methods has gradually become available. Artificial neural networks have been shown to be universal approximators [20]. A formal framework has been developed for modeling by RBFNs by showing their relationship to regularization methods in approximation theory [21]. This framework has been exploited to develop generalized forms of radial basis function networks such as hyper basis functions and regularization networks [22]. Among statistical methods, linear methods have been subjected to significant theoretical analysis, and the connections between linear statistical methods including OLS, PCR, PLS, and RR have been studied [23–25]. The universal approximation property of certain classes of nonlinear statistical methods is also proved [26]. PPR has been analyzed theoretically [27,28], to show that the PPR model is invariant under rotation and scaling, and many classes of functions can be modeled by it.

Greater understanding of the properties of empirical modeling methods has also led to some cross-fertilization between various methods. For example, OLS, PLS and PCR have been combined [24]; PCA is used for initializing the input edge weights of a BPN [29]; ANN are used for implementing PCA and PLS [30,31]; PCA and PLS are used for reducing the dimensionality of the input space before modeling by MARS [32]; BPN are used to determine the nonlinear inner relationship in nonlinear PLS [33], as well as NLPCA [34]. The benefits of these methods indicate that similar hybridization of other properties of different empirical modeling may be desirable. Such a combination of empirical modeling methods requires deep insight into the properties, similarities and differences between different methods.

In this paper, we present a common framework for bringing out the similarities and differences between

empirical modeling methods and enabling greater understanding of their properties. This framework is based on the insight that the model determined by all empirical modeling methods may be represented as a weighted sum of basis functions, and that different methods may be derived depending on decisions about the nature of the input transformation, the type of activation functions, and the optimization criteria for determining the adjustable parameters. Transformation of the inputs is essential for fighting the curse of dimensionality in empirical modeling. All empirical modeling methods use one of the three techniques for input transformation: methods based on linear projection combine the inputs by projection on a linear hyperplane as a linear weighted sum, methods based on nonlinear projection combine the inputs by projection on a nonlinear hypersurface, and partition-based methods divide the input space by selecting only those inputs that are most relevant to determining the best empirical model. The shape of the activation functions may be fixed independent of the available data, or may adapt to the data. Fixed-shape activation functions include linear, sigmoid and Gaussian functions, whereas adaptive-shape activation functions are determined by univariate smoothing techniques such as regression splines, variable span smoothers, and back-propagation networks. Finally, the optimization criteria for determining the input transformation parameters may involve either the input space only, or the input and output space, or the output space only. The common framework developed in this paper allows easy comparison of the properties of various empirical modeling methods. Such insight may be used to guide the selection of the best method for a given empirical modeling task, and to develop novel techniques that unify various existing empirical modeling methods, as demonstrated by the unification of methods that combine inputs by linear projection [35]. This paper complements several recent overviews and tutorials on empirical modeling methods [36–40]. None of these papers present a simple framework for comparing all empirical modeling methods and for combining the features of existing methods, as described in this paper.

The rest of this paper is organized as follows. The common framework that brings out the relationships, differences, and similarities between various empirical modeling methods is presented in Section 2. This

framework is used to provide an overview of existing empirical modeling methods in Section 3. This section consists of three major subsections divided according to the nature of the input transformation. Methods based on linear projection include OLS, PLS, PCR, RR, BPN with one hidden layer, PPR, NLPCR, and NLPLS. Methods based on nonlinear projection include BPN with multiple hidden layers, RBFN, NLPCA and NLPLS based on nonlinear input transformation, and partition-based methods include CART or inductive decision trees, and MARS. A simple example is then solved to further illustrate the properties of various empirical modeling techniques and provide additional insight in Section 4. The challenges for developing techniques that unify existing empirical modeling methods and combine their properties are discussed in Section 5.

2. Common framework for comparing and combining empirical modeling methods

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

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

where \hat{y}_k is the k th predicted output or response variable, θ_m the m th basis function, β_{mk} the output weight or regression coefficient relating the m th basis function to the k th output, α the matrix of basis function parameters, ϕ_m represents the input transformation, and x_1, \dots, x_J are the inputs or predictor variables. The latent variables, or transformed inputs are represented as

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

The model given by Eq. (1) may also be represented as an artificial neural network, where α are the edge weights of the input and hidden layers, β the edge weights of the output layer, and ϕ and θ are the basis functions. Specific empirical modeling methods may be derived from Eq. (1) depending on decisions about the nature of input transformation, type of activation functions, and optimization criteria. These decisions form the basis of the common framework developed in

this paper for comparing all empirical modeling methods, and are described in the rest of this section.

2.1. Nature of input transformation

Empirical modeling problems often involve a large number of measured variables as inputs. Reducing the dimensionality of the input space is essential since 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 combining the inputs to *latent* variables that compress the inputs by capturing their relation with less latent variables than the number of inputs. Such dimensionality reduction is accomplished by extracting the relationship among inputs, or distribution of measured data, or relevance of input variables for predicting the output. Thus, empirical modeling methods may be categorized as follows depending on the nature of input transformation.

- *Methods based on linear projection* project the inputs on a linear hyperplane, as shown in Fig. 1(a), before applying the basis function. This exploits the linear relationship among input variables by combining them as a linear weighted sum to form the latent variables.

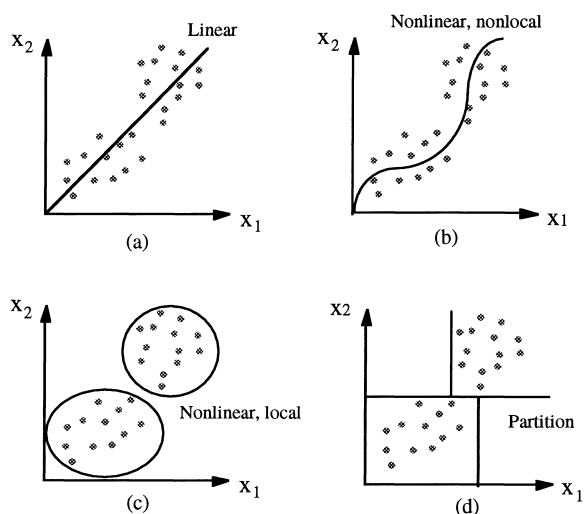


Fig. 1. Input transformation in: (a) methods based on linear projection; (b) methods based on nonlinear projection, non-local transformation; (c) methods on nonlinear projection, local transformation; (d) partition-based methods.

- *Methods based on nonlinear projection* project the inputs on a nonlinear hypersurface resulting in latent variables that are nonlinear functions of the inputs, as shown in Fig. 1(b) and (c). If the inputs are projected on a localized hypersurface such as a hypersphere or hyperellipse, then the basis functions are local, as depicted in Fig. 1(c). Otherwise, the basis functions are non-local in nature. This approach exploits the nonlinear relationship between the inputs.
- *Partition-based methods* reduce 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, as depicted in Fig. 1(d).

The nature of input transformation provides a convenient criterion for categorizing empirical modeling methods, and forms the basis of the overview presented in Section 3.

2.2. Type of activation functions

The function that relates the latent variable to the output, $\theta(z)$ is two-dimensional in nature, and is referred to as the activation function, while the function that relates the inputs to the output, $\theta(\alpha; x)$ is referred to as the basis function. The wide variety of activation functions used in empirical modeling methods may be broadly divided into the following two categories:

- *Fixed-shape activation functions.* The activation functions in several empirical modeling methods are of a fixed shape such as linear, sigmoid, Gaussian, wavelet, or sinusoid. Adjusting the activation function parameters changes their location, and size, but their shape is decided a priori, and remains fixed.
- *Adaptive-shape activation functions.* Some empirical modeling methods relax the fixed-shape requirement and allow the activation functions to adapt their shape, in addition to their location and size, 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. Adap-

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

Some common examples of fixed- and adaptive-shape activation functions are shown in Fig. 2. Column I represents various types of activation functions with the x -axis representing the transformed inputs or latent variable, z , and the y -axis representing the output, y . The multi-dimensional character of the basis functions in the input–output space is shown in Column III and is determined by the nature of the input transformation depicted in Column II. For example, if the input transformation is nonlocal, then the multi-dimensional basis function in the input–output space is also nonlocal in the direction of the input transformation. This is depicted by the basis functions shown in Fig. 2(a)–(c) and (e). The basis functions obtained when the inputs are transformed by linear projection are also known as ridge functions. However, if the input transformation is local, then the basis functions in the input–output space are also localized, as depicted by the two-dimensional Gaussian shown in Fig. 2(d). An example of an adaptive shape activation function with input transformation by linear projection is shown in Fig. 2(e). A typical basis function for partition-based methods is shown in Fig. 2(f). This basis function is formed by multiplication of the Heaviside functions shown in Column I that partition the input space as shown in Column II, to result in the multidimensional basis function shown in Column III. Nonlinear empirical modeling methods are universal approximators if the basis function in the input–output space can completely span the relevant functional space.

2.3. Optimization criteria

The aim of any empirical modeling method is to extract the underlying input–output relationship and/or input transformation from the available data. 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

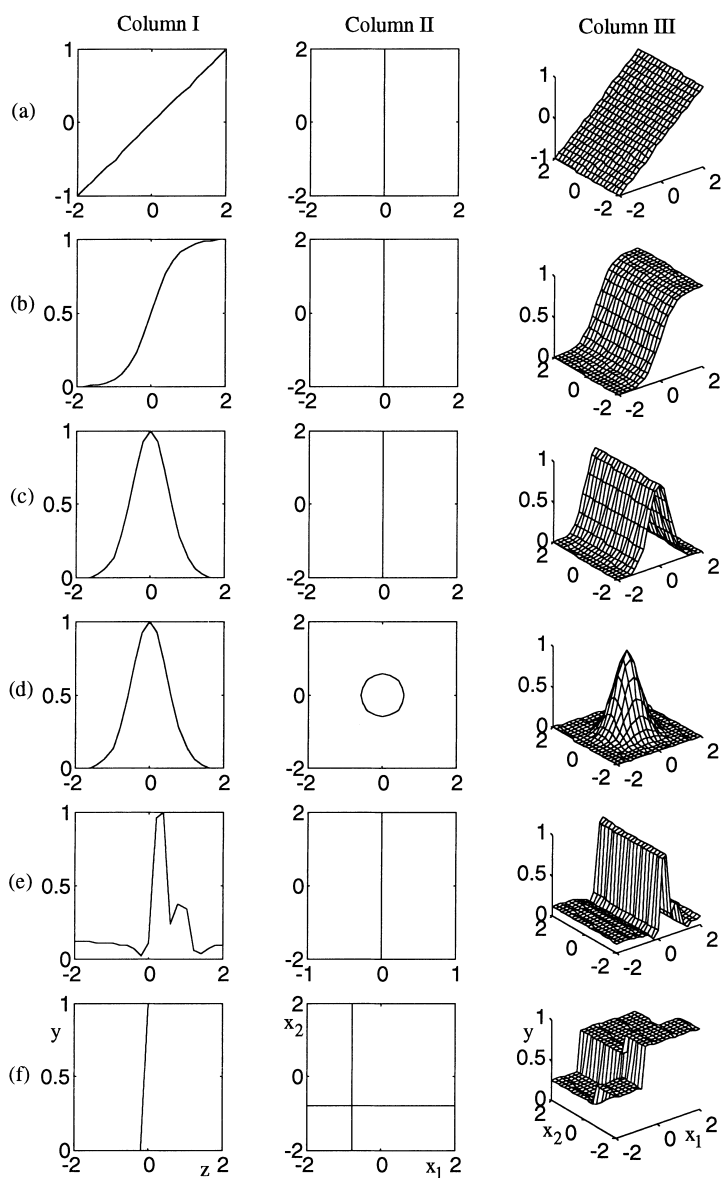


Fig. 2. Types of activation functions in empirical modeling methods. Column I: activation function in transformed input, z , vs., output, y , space; Column II: nature of input transformation; Column III: basis function in input–output space: (a) linear methods based on linear projection; (b) sigmoid in BPN; (c) Gaussian in methods based on linear projection; (d) Gaussian activation function in local methods; (e) ridge function in PPR; (f) piece-wise linear function in CART.

for determining the input transformation, and the relationship between the transformed input and outputs. This separation of the empirical modeling optimization criteria provides explicit control over the dimensionality reduction by input transformation, and

often results in more accurate empirical models as discussed in Section 3. Empirical modeling methods may be divided into two categories depending on whether the optimization criterion for the input transformation contains information from:

Table 1
Comparison matrix for empirical modeling methods

Method	Input transformation	Basis function	Optimization criteria
OLS	Linear projection	Fixed shape, linear	α – max. squared correlation between projected inputs output β – min. output prediction error
PLS	Linear projection	Fixed shape, linear	α – max. covariance between projected inputs and outputs 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	$[\alpha, \beta]$ – min. output prediction error
NLPCA	Nonlinear projection, nonlocal	Adaptive shape	$[\alpha, \phi]$ – min. output prediction error
RBFN	Nonlinear projection, local	Fixed shape, radial	$[\sigma, t]$ – min. distance between inputs and cluster center β – min. output prediction error
CART	Input partition	Adaptive shape, piecewise constant	$[\beta, t]$ – min. output prediction error
MARS	Input partition	Adaptive shape, spline	$[\beta, t]$ – min. output prediction error

- *only the inputs*, for example, when maximizing the variance of the measured data captured by the transformed inputs, as in PCA,
- *both inputs and outputs*, for example, when maximizing the covariance between the transformed inputs and outputs, as in PLS, or when minimizing the output prediction error, as in OLS.

The optimization criterion for determining the output parameters and basis functions is to minimize the output prediction error, and is common to all empirical modeling methods.

The nature of the input transformation, type of activation 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. Furthermore, this framework indicates that various empirical modeling methods may be combined by developing methods that unify the different approaches for transforming the inputs, determining the shape of the activation functions, or the optimization criteria. Indeed, such an approach has been developed for unifying methods based on linear projection [35].

3. Overview of empirical modeling methods

Based on the common comparison framework presented in Section 2, empirical modeling may be defined as the following approximation problem.

Empirical modeling problem. Let $\hat{x}(\phi(\alpha, x))$ and $\hat{y}(\beta, \theta(\alpha, x))$ be real-valued approximation functions depending continuously on $x \in X$ and on parameters, α , β , and function, $\theta(\alpha, x)$. Given the distance functions ρ_1 and ρ_2 , determine $\alpha^* \in \mathcal{R}$, $\phi^*(\alpha^*, x) \in \mathcal{L}^p$, such that,

$$\rho_1[\hat{x}(\phi(\alpha^*, x)), x, \hat{y}(\beta^*, \theta^*(\alpha^*, x)), y(x)] \leq \rho_1[\hat{x}(\phi(\alpha, x)), x, \hat{y}(\beta, \theta(\alpha, x)), y(x)], \quad (2)$$

and determine $\beta^* \in \mathcal{R}$, $\theta^*(\alpha^*, x) \in \mathcal{L}^p$, such that,

$$\rho_2[\hat{y}(\beta^*, \theta^*(\alpha^*, x)), y(x)] \leq \rho_2[\hat{y}(\beta, \theta(\alpha, x)), y(x)], \quad (3)$$

where ρ_1 and ρ_2 are functions of the distance between the actual and approximated functions.

The definition of the empirical modeling problem given above incorporates approximation of the input and the output space. This definition is different from the conventional definition of approximation problems [41] and definitions given for ANN modeling [21,42], where the entire emphasis is on minimizing the error of approximation of the outputs only. This broader definition allows inclusion of various neural

and statistical methods for empirical modeling. Separate objective functions, ρ_1 for the input transformation parameters, and ρ_2 for the basis functions and regression parameters allow empirical modeling methods to explicitly trade-off the emphasis on capturing the relationship between the input variables, and minimizing the output error of approximation. For all methods, the function, ρ_2 focuses only on minimizing the output error of approximation, whereas ρ_1 may also focus on capturing the relationship between the inputs. Different empirical modeling methods are obtained depending on the emphasis of ρ_1 on approximating the inputs. The benefits of satisfying these two separate objective functions may be explained by their effect on the mean-squares error of approximation.

The theoretical mean-squares error of approximation between the actual output, y , and approximated output, \hat{y} , is equal to the sum of the variance of the predicted output and the squared bias of the predicted output from the theoretical value [7,43],

$$E(y - \hat{y})^2 = E(\hat{y} - E(\hat{y}))^2 + (E(\hat{y}) - y)^2,$$

that is,

$$\text{MSE}(\hat{y}) = \text{Variance}(\hat{y}) + (\text{Bias}(\hat{y}))^2. \quad (4)$$

If the selected form of the model matches that of the system being modeled, and both objective functions minimize the output prediction error, then the resulting model is unbiased. In many modeling situations, such as when the amount of available training data is small, it is not possible to obtain a small mean-squared error between the actual output and the model prediction. In such a case, if the model is unbiased, then the model variance will be as large as the mean-squared error. Since the variance represents the degree of uncertainty of the model, a large variance will result in poor model performance on previously unseen data. For such a problem, a biased modeling technique will result in a smaller variance, leading to improved model generality and better model performance on previously unseen data.

Separation of the objective functions in empirical modeling provides control over the extent of model bias. For example, for unbiased modeling methods, both ρ_1 and ρ_2 minimize the output prediction error. For linear methods, the bias may be maximized if ρ_1 minimizes the input approximation error, and ρ_2 mini-

mizes the output prediction error, as is the case with PCR. The model bias may be decreased by reducing the emphasis on capturing the input relationship, and increasing the emphasis on minimizing the output prediction error in ρ_1 . The model bias is also determined by the number of basis functions in the model. Increasing the number of basis functions decreased the model bias, but increases the model variance, resulting in a minimum mean-squares error where the bias-variance trade-off is optimized.

In the rest of this section, the comparison framework presented in Section 2 and the definition of the empirical modeling problem are used to provide an overview and insight into some of the popular empirical modeling methods. The discussion focuses primarily on modeling with multiple inputs and a single output. Modifications to modeling with multiple outputs are discussed where appropriate. The empirical modeling methods are separated into three categories according to the nature of the input transformation.

3.1. Methods based on linear projection

These methods project the inputs on a hyperplane by combining them as a linear weighted sum, and are among the most widely used empirical modeling methods for linear and nonlinear modeling. In general, the model determined by methods based on linear projection for multiple-input–single-output problems may be represented by specializing Eq. (1) to:

$$\hat{y} = \sum_{m=1}^M \beta_m \theta_m \left(\sum_{j=1}^J \alpha_{jm} x_j \right), \quad (5)$$

where α_{jm} is the input weight relating the j th input, x_j , to the m th basis function, θ_m . This model may also be represented as a neural network with the α_{jm} representing the input edge weight, θ_m representing the activation functions in the single hidden layer, and β_m representing the outer edge weights. The input weights correspond to the direction cosines of the hyperplane on which the inputs are projected, and determine its orientation. This linear combination of inputs constitutes the latent variables extracted from the input space, and is given by

$$z_m = \sum_{j=1}^J \alpha_{jm} x_j \quad (6)$$

$$\mathbf{Z} = \mathbf{X}\alpha,$$

where \mathbf{Z} is an $I \times M$ matrix of projected values or latent variables, and \mathbf{X} is an $I \times J$ matrix of measured inputs. The inputs may be reconstructed from the latent variables, z_m , and projection directions α_{jm} by inverting Eq. (6). If the input dimensionality is reduced by eliminating some latent variables and corresponding projection directions, then the reconstruction will approximate the original inputs.

The basis functions in methods based on linear projection are ridge functions in the input–output space, as illustrated in Fig. 2(a)–(c) and (e). The distribution of the data in the transformed input–output space depends on the orientation of the projection hyperplane. For example, in Fig. 2(a)–(c) and (e), projection of the output variable on the plane perpendicular to that in Column II results in the smooth curve in the latent variable–output space as shown in Column I. Any other orientation of the plane increases the scatter of the projected data.

3.1.1. Linear methods

Restricting the basis functions to be linear results in linear input–output models of the following general form:

$$\hat{y} = \sum_{m=1}^M \beta_m \sum_{j=1}^J \alpha_{jm} x_j. \quad (7)$$

These are among the simplest empirical modeling methods and are used extensively in process monitoring and control. A brief comparison of these methods is provided below. Detailed descriptions are available in several books and insightful papers [7,24,25,44]. For all these methods, the regression coefficients are computed in the same manner by minimizing the output mean-squares error. The objective functions used to optimize the projection directions or input weights, however, are different, as summarized in Table 1.

Ordinary least squares regression (OLS). The model determined by OLS is represented as

$$\hat{y} = \sum_{j=1}^J b_j x_j. \quad (8)$$

Given several observations, Eq. (8) constitutes a system of linear equations, $\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}$, where \mathbf{b} is the $J \times 1$ vector of unknown parameters. The value of \mathbf{b} is

determined by minimizing the mean-squares error of approximation

$$\max_{\mathbf{b}} (\|\mathbf{y} - \hat{\mathbf{y}}\|^2),$$

where the notation $\|\cdot\|^2$ indicates the sum of the squares of each component in the argument. Usually, the number of observations, I , is more than the number of input variables, J , and the solution to the least-squares approximation problem is given by the product of the pseudo inverse of \mathbf{X} and \mathbf{y} . This solution does not consider the relationship between the inputs in the optimization, and is the best linear unbiased estimate.

The OLS model may be included in the comparison framework by representing it as a special case of Eq. (7) with only one linear basis function as

$$\hat{y} = \beta_1 \sum_{j=1}^J \alpha_{j1} x_j.$$

This model involves linear regression of the output on the projected inputs, $\mathbf{X}\alpha_1$. The projection directions or basis function parameters, α_1 are computed to maximize the squared correlation between the actual output and the projected inputs as [24]

$$\max_{\alpha_1} \{ \text{corr}^2(\mathbf{y}, \mathbf{X}\alpha_1) \}. \quad (9)$$

The regression parameter, β_1 is computed to minimize the mean-square error of approximation given by the pseudo inverse of $\mathbf{X}\alpha_1$. The optimization criterion given by Eq. (9) is equivalent to minimizing the output mean-squares error of approximation [35]. The regression coefficient, β_1 is not the same as the regression coefficient, \mathbf{b} , in Eq. (8) since β_1 is the linear regression coefficient of \mathbf{y} on $\mathbf{X}\alpha_1$, whereas \mathbf{b} is the linear regression coefficient of \mathbf{y} on \mathbf{X} . The product of β_1 and α_1 is then equal to \mathbf{b} . For modeling problems with multiple outputs, OLS minimizes the sum of the mean-squares error of each output. Thus, the matrix of OLS parameters is given by the pseudo-inverse of the inputs multiplied by the matrix of outputs.

If the inputs are correlated, then the pseudo inverse of \mathbf{X} , may not exist and the OLS coefficients cannot be computed. Even with partially correlated inputs, the covariance matrix can be nearly singular and possess very small eigenvalues, making the OLS solution extremely sensitive to small changes in the measured

data. The unbiased nature of the OLS model may also result in a large model variance and error on testing data. In such cases, OLS may not be appropriate for empirical modeling, and biased methods where the latent variables attempt to capture the relationship between the inputs may perform better.

Principal component analysis. PCA is an input transformation method that extracts projection directions, or principal component loadings by satisfying the following optimization criterion:

$$\max_{\alpha_m} \{ \text{var}(\mathbf{X}\alpha_m) \}. \quad (10)$$

The projection directions are constrained to be orthonormal, and are eigenvectors of the input covariance matrix, $\mathbf{X}^T\mathbf{X}$. The dimensionality of the input space may be decreased by selecting a subset of the latent variables that capture most of the variance in the measured data. The data matrix may then be approximated as

$$\hat{\mathbf{X}} = \hat{\mathbf{Z}}\hat{\boldsymbol{\alpha}}^T, \quad (11)$$

where $\hat{\mathbf{Z}}$ denotes the matrix of selected latent variables, and $\hat{\boldsymbol{\alpha}}$ are the selected orthonormal projection directions. The PCA optimization criterion given by Eq. (10) is equivalent to minimizing the mean-squares error between the actual and approximated inputs [45]

$$\min_{\alpha_m} \left[\sum_j (\|\mathbf{x}_j - \hat{\mathbf{x}}_j\|^2) \right]. \quad (12)$$

All the projection directions may be computed simultaneously via eigenvalue decomposition of the input covariance matrix, or hierarchically by the non-linear iterative partial least squares (NIPALS) algorithm [7]. This algorithm determines one projection direction at a time that captures the maximum residual variance of the inputs, while requiring orthogonal directions.

Principal component regression. PCR extracts the latent variables from the input space by PCA, and then performs OLS regression between the selected latent variables and output. The model determined by PCR may also be expressed by Eq. (7), where $\mathbf{X}\alpha_m$ are the principal component scores, α_m are the principal component loadings or projection directions, and β_m are the regression coefficients. Unlike OLS, inversion of the covariance matrix of the principal component scores to find the regression coefficients in PCR is

possible even when the inputs are highly correlated, since the principal component loadings are mutually orthogonal and uncorrelated. PCR considers only the input space in finding the projection directions while ignoring the input–output relationship. Consequently, the PCR model is biased, and is best suited for problems where the ratio of training data to input variables is small.

Partial least squares regression. The PLS algorithm was developed as a compromise between OLS and PCR for problems with correlated data and small ratio of training data to inputs [46]. The resulting model is similar to PCR and is also represented by Eq. (7). However, in PLS the projection directions are computed based on the characteristics of both inputs and outputs. Unlike PCR, which computes the loadings to maximize the objective function involving only the inputs, PLS computes the loadings by the following objective function:

$$\max_{\alpha_m} \{ \text{corr}^2(\mathbf{y}, \mathbf{X}\alpha_m) \text{var}(\mathbf{X}\alpha_m) \}, \quad (13)$$

while constraining the loadings, $\boldsymbol{\alpha}$, or scores, $\mathbf{Z}=\mathbf{X}\boldsymbol{\alpha}$, to be orthogonal. The PLS objective function may also be represented as minimizing the distance between the projected inputs and output [47]. Thus, the PLS objective function for computing the projection direction is a combination of the OLS and PCR objective functions. The resulting projection directions are rotated away from the PCR projection directions towards the OLS projection directions in the input space with a bias between that of OLS and PCR. The PLS projection directions may be computed simultaneously as the eigenvectors of the $(\mathbf{Y}^T\mathbf{X})^T(\mathbf{Y}^T\mathbf{X})$ matrix, or hierarchically by the NIPALS method. For problems with multiple outputs, PLS projects the outputs on a linear hyperplane and seeks inputs and output projection directions that maximize the covariance between the projected inputs and projected outputs.

Ridge regression. This is another approach for introducing bias and avoiding problems due to correlated inputs in OLS modeling. The RR model is usually written as Eq. (8) with the OLS objective function modified to minimization of a penalized mean-squares error,

$$\min_{\mathbf{b}} (\|\mathbf{y} - \hat{\mathbf{y}}\|^2 + \lambda \mathbf{b}^T \mathbf{b}). \quad (14)$$

The solution to Eq. (14) is

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}.$$

Selecting $\lambda=0$ provides the OLS solution, while $\lambda=\infty$ gives the mean value of the output. The objective function for computing the projection directions for RR may also be cast into the framework of Eq. (7) as [25]

$$\max_{\alpha_1} \left\{ \text{corr}^2(\mathbf{y}, \mathbf{X}\alpha_1) \frac{\text{var}(\mathbf{X}\alpha_1)}{\text{var}(\mathbf{X}\alpha_1) + \lambda} \right\}. \quad (15)$$

Eq. (14)[15] show that $\lambda \neq 0$ moves the projection directions away from those of OLS, by capturing more of the variance in the input space, increasing the bias. As usual, the regression parameters, β , are computed by minimizing the mean-squares error of approximation.

3.1.2. Nonlinear methods

The use of nonlinear activation functions in methods based on linear projection results in nonlinear empirical models. These methods include nonlinear PCR, nonlinear PLS, PPR and BPN with one hidden layer, and their model is also given by Eq. (5). The nonlinear activation functions may be of fixed or adaptive shape. If adaptive, the shape of the activations is also determined via optimization.

Backpropagation networks. The activation functions in BPN are nonlinear and of a fixed shape, usually sigmoid. The mathematical representation of a BPN is given by

$$\hat{y} = \bar{y} + \sum_{m=1}^M \beta_m \sigma_m \left(\sum_{j=1}^J \alpha_{jm} x_j - \alpha_{0m} \right), \quad (16)$$

where \bar{y} and α_{0m} are the bias terms of the output and hidden layers, respectively. Eq. (16) may be reduced to the form of Eq. (5) by using a dummy variable, $x_0 = -1$, and normalizing the outputs resulting in

$$\hat{y} = \sum_{m=1}^M \beta_m \sigma_m \left(\sum_{j=0}^J \alpha_{jm} x_j \right).$$

The model parameters in a BPN are computed to satisfy the output mean-squares error of approximation

$$\min_{\alpha, \beta} \left\{ \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \right\}. \quad (17)$$

This objective function is identical to that used by OLS, and finds projection directions that minimize the output prediction error while ignoring any relationship among the input variables.

Methods for training BPN have received considerable attention. The most common training methodology optimizes the input weights (projection directions) and output weights (regression coefficients) simultaneously for the entire network via the error back propagation algorithm [48]. This procedure can be computationally expensive since the number of nodes is determined by training separate networks with different number of nodes. Hierarchical training methods such as cascade correlation [49] have also been developed for faster node-by-node training.

Projection pursuit regression. Projection pursuit regression (PPR) is a nonlinear multivariate statistical modeling technique developed by Friedman and Stuetzle [50] for analyzing high-dimensional data. The PPR model is similar to BPN, except that the basis functions adapt their shape to the available training data. The input and output parameters and shape of the basis functions are determined to minimize the mean-squares error of approximation,

$$\min_{\alpha_m, \beta_m, \theta_m} \left\{ \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \right\}. \quad (18)$$

The training procedure for PPR builds the empirical model in a hierarchical or node-by-node manner. The adjustable parameters and basis function for each node are determined by minimizing the output residual error of approximation that has not been captured by previously introduced nodes in the model. The parameters and basis functions of previously added nodes are then readjusted to optimize the entire model. Various techniques have been suggested for determining the basis functions including, variable span smoothers [51], splines [52] and Hermite polynomials [53]. The adaptive shape of the basis functions usually results in models with less nodes than models determined by techniques with basis functions of fixed shape such as BPN. The PPR algorithm has been extended for multiple-input–multiple-output modeling by a technique called smooth multiple additive regression technique (SMART) [51]. The outputs are also projected on a linear hyperplane, with the projection directions supplied by the user.

The parameters and basis functions for the multi-input–multi-output model are also determined by satisfying Eq. (18).

Nonlinear principal components regression. NLPCR extends linear PCR to nonlinear modeling by using nonlinear basis functions. Various types of adaptive shape basis functions have been suggested, including polynomials [54] and variable span smoothers [55]. The objective function for determining the projection directions in NLPCR is the same as that of linear PCR given by Eq. (10). Since the basis functions are not a part of the objective function, PCR and NLPCR result in identical projection directions. As in PPR, using adaptive-shape basis functions provides the flexibility to find the smooth basis functions that best capture the structure of the hypersurface being approximated. The resulting algorithm is similar to the linear PCR algorithm except for an additional step for determining the nonlinear basis functions.

Nonlinear partial least squares. Extending linear PLS to NLPLS is analogous to extending linear PCR to NLPCR. The optimization criterion for determining the projection directions by NLPLS lies between that used by PPR and NLPCR, and is a function of the inputs and output. Several variations of the linear PLS optimization criterion have been suggested for determining the projection directions, depending on whether the nonlinear basis functions are included in the optimization. The NLPLS algorithms in [33,56] use the linear PLS optimization criterion given by Eq. (13) to determine the projection directions. The nonlinear activation functions need to be determined only once for each latent variable. If the nonlinearity of the basis functions is incorporated in the objective function for determining the projection directions, the quality of the empirical model usually improves, but the computational complexity increases [16,31,57]. The objective function is then modified to

$$\max_{\alpha_m} \{ \text{cov}(\mathbf{y}, \theta_m(\mathbf{X}\alpha_m)) \}. \quad (19)$$

For linear basis functions, Eq. (19) reduces to Eq. (13). The basis functions, or the so-called inner relation in NLPLS, may be determined by a variety of nonlinear smoothing techniques including quadratic functions [57], variable span smoothers [56], back-propagation networks [31,33], and splines [32].

3.2. Methods based on nonlinear projection

The model determined by methods based on nonlinear projection may be represented as Eq. (1), where $\phi(\boldsymbol{\alpha}; x_1, x_2, \dots, x_J)$ represents the nonlinear input transformation, and $\boldsymbol{\alpha}$ is the input transformation parameter. The nonlinearly transformed inputs constitute the nonlinear latent variables extracted from the input space,

$$z_m = \phi_m(\boldsymbol{\alpha}; x_1, \dots, x_J). \quad (20)$$

The original inputs may be reconstructed from the latent variables and input transformation parameters by determining the inverse of Eq. (20). If some latent variables are ignored, then the inputs may be approximated as

$$\hat{\mathbf{X}} = \zeta(\hat{\boldsymbol{\alpha}}, \hat{\mathbf{Z}}), \quad (21)$$

where ζ represents the inverse of the function, ϕ , and may have to be determined empirically due to the nonlinear relationship.

Depending on the nature of the input transformation, methods based on nonlinear projection may be further divided into local and non-local methods. Local methods divide the input space into localized or closed regions by projecting the inputs on hyperspheres or hyperellipses, whereas nonlocal methods project the inputs on a nonlinear hypersurface that is unbounded or global along at least one direction. As in methods based on linear projection, the basis functions in methods based on nonlinear projection may also be of fixed or adaptive shape. The basis functions plotted against the latent variables are univariate, but need not be ridge functions in the input–output space, as depicted in Fig. 2(d).

3.2.1. Local methods

Local empirical modeling methods are among the most widely used methods based on nonlinear projection, and include nearest neighbor clustering methods, adaptive resonance theory (ART), RBFNs, and wavelet networks (wave-nets). The localized basis functions do not permit these models to extrapolate much beyond the region where training data are available. Furthermore, model adaptation with new data can be more efficient since all the basis functions need not be updated. These properties have made local methods

very popular for classification problems such as pattern recognition, spectral analysis and fault diagnosis [11,58].

Nearest neighbors clustering. This input transformation approach finds a set of centers that minimize the Euclidean distance between each data point and the cluster center:

$$\min_i \left\{ \sum_{m=1}^M \sum_{i=1}^I B_{mi} \| \mathbf{x}_i - \mathbf{t}_m \|^2 \right\}, \quad (22)$$

where \mathbf{x}_i is a vector of the i th measurement, M the desired number of clusters, and B_{mi} is an $M \times I$ matrix of the cluster partition or membership function. The hypersphere around each cluster center is then determined to ensure sufficient overlap between the clusters for a smooth fit by criteria such as the P -nearest neighbor heuristic,

$$\sigma_k \left[\frac{1}{P} \sum_{i=1}^P \| \mathbf{x}_i - \mathbf{t}_m \|^2 \right]^{1/2}, \quad (23)$$

where \mathbf{x}_i are the P -nearest neighbors of the centers, \mathbf{t}_m , resulting in hyperspherical receptive fields. The objective functions for both k -means clustering and the P -nearest neighbor heuristic given by Eq. (22)[23] use information only from the inputs. Consequently, the clustering process is similar to PCA, but with inputs projected on a hypersphere in clustering instead of on a hyperplane in PCA.

Adaptive resonance theory. ART is a biologically inspired clustering technique where the hyperspheres are of an identical size determined by a threshold, called the vigilance value [58,59]. The clustering is performed by minimizing a selected distance measure between the data and clusters. A variety of metrics may be used including a weighted linear combination of inputs, or Euclidean distance given as Eq. (22). The iterative nature of the ART training methodology and the fixed size of each hypersphere permit adaptation of the clusters as more data are obtained. Several extensions of ART have been developed to deal with discrete or analog inputs, as well as for input–output modeling [60].

Radial basis function network. The basis functions in RBFNs are of the form, $\theta(\|x - t_m\|^2)$, where t_m denotes the center of the basis function. One of the

most popular RBFs is the Gaussian,

$$\theta_m = \exp \left(- \frac{(x_i - t_m)^2}{\sigma_m^2} \right),$$

where, the parameter, σ_m determines the extent of localization of the basis function. Multidimensional Gaussians may be obtained by multiplying univariate Gaussians resulting in an empirical model of the following form:

$$\hat{y} = \sum_{m=1}^M \beta_m \exp \left(- \sum_{j=1}^J \frac{1}{\sigma_{jm}^2} (x_j - t_{jm})^2 \right). \quad (24)$$

The model represented by Eq. (24) is similar to Eq. (5) for methods based on linear projection, but with inputs transformed by translation and quadratic operation, and exponential basis functions.

The most popular method for modeling by RBFN involves separate steps for determining the basis function parameters, σ_{jm} and t_{jm} , and the regression coefficients, β_m . The basis function parameters are determined without considering the behavior of the outputs by k -means clustering and the P -nearest neighbors heuristic. The regression parameters are then determined to minimize the output mean-squares error of approximation. Thus, the optimization criterion for determining the input transformation is given by Eq. (22), and considers the input space only, making RBFNs trained by this method analogous to PCR. Due to the known disadvantages of computing the basis function parameters based on the input space only, various approaches have been suggested for incorporating information about the output error for determining the basis function parameters [61]. Hierarchical methods have also been developed for modeling in a step-wise or node-by-node manner using Gaussian basis functions [62] and wavelets [42]. Techniques have also been developed to project the inputs on hyperellipses, instead of hyperspheres. These ellipsoidal basis function networks allow non-unity and unequal input weights, except zero and negative values, causing elongation and contraction of the spherical receptive fields into ellipsoidal receptive fields [63]. RBFNs have also received significant theoretical attention by Poggio and Girosi [21] who show that RBFN modeling is equivalent to solving the regularization problem by minimizing the

objective function

$$\min(\|\mathbf{y} - \hat{\mathbf{y}}\|^2) + (\lambda \|P\hat{\mathbf{y}}\|^2),$$

where λ is an adjustable parameter that trades-off the interpolation versus the generalization of $\hat{\mathbf{y}}$, and P is a stabilizing operator. Different types of radial basis functions are obtained depending on the nature of the operator, P .

3.2.2. Nonlocal methods

BPN with multiple hidden layers. A BPN is not required to have more than a single hidden layer to satisfy the universal approximation property, but the model with one hidden layer always combines the inputs by linear projection. Some practitioners prefer using BPN with multiple hidden layers to allow a nonlinear combination of the input variables before transformation by the activation function. The nonlinear projection may be local or global in the input space. The resulting empirical model is a specialization of Eq. (1) to

$$\hat{\mathbf{y}} = \sum_{m=1}^M \beta_m \sigma_m \left(\sum_{j_h=0}^{J_h} \alpha_{mj_h} \sigma_{j_h} \left(\sum_{j_{h-1}=0}^{J_{h-1}} \alpha_{mj_{h-1}} \sigma_{j_{h-1}} \left(\dots \sigma_1 \left(\sum_{j_1=0}^{J_1} \alpha_{mj_1} x_{j_1} \right) \right) \right) \right),$$

where J_h denotes the number of nodes in the h th hidden layer, and J_1 is equal to J which is the number of inputs. Each node projects the inputs on a hyperplane. Thus, the first hidden layer projects the inputs on a combination of linear hyperplanes as shown in Fig. 3(a). Each node in the second hidden layer projects the outputs from the first layer on another linear hyperplane resulting in projection of the inputs on a convex hypersurface built by combining linear hyperplanes, as shown in Fig. 3(b). The number of edges in the nonlinear projection hypersurface is determined by the number of nodes in the hidden layer.

Introducing a third hidden layer combines the closed convex regions to generate arbitrarily shaped hypersurfaces in the input space, as depicted in Fig. 3(c) [64].

The training methodology for BPN with multiple hidden layers is not significantly different from that for BPN with a single hidden layer, and all the unknown parameters are optimized jointly via the

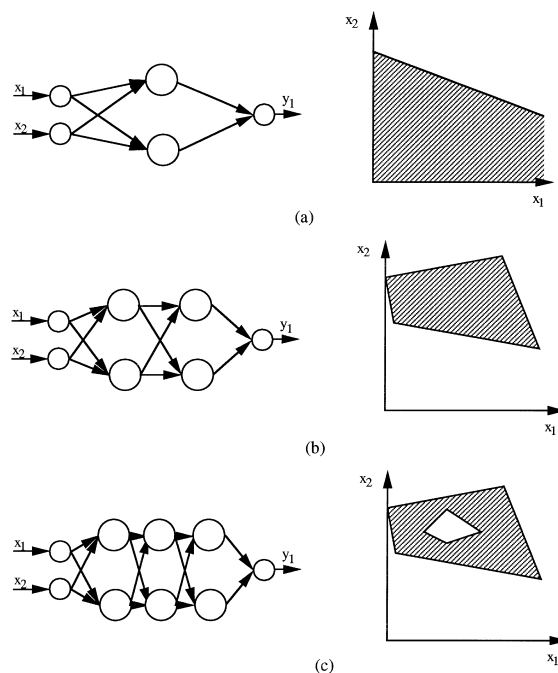


Fig. 3. Nature of projection surface for each node in the last hidden layer of BPN with multiple hidden layers: (a) single hidden layer projects input on a plane; (b) two hidden layers project inputs on a convex region; (c) three hidden layers project inputs on convex or concave region. Number of edges in projection surface are determined by number of nodes in hidden layers other than the last one [64].

error backpropagation algorithm for the entire network. The objective function being minimized is the mean-squares error between the actual and the predicted output as given by Eq. (17), without taking the relationship between the inputs into account.

Nonlinear PCA. Encouraged by the ability of PCA to transform the input variables to useful latent variables, several techniques have been developed for extracting nonlinear relationships among the variables. NLPCA is an extension of linear PCA, where the input variables are transformed nonlinearly to maximize the variance captured by each nonlinear principal component. Thus, the objective function for NLPCA is to find the nonlinear combination of inputs that maximizes the variance of the projected inputs,

$$\max_{\alpha_m, \phi_m} \{\text{var}(\phi(\alpha_m, \mathbf{X}))\}.$$

This objective function is equivalent to minimizing the error between the available data and its approximation [45], and may be written as Eq. (12) for determining the optimum values of the function, ϕ_m , and its parameters, α_m , with $\hat{\mathbf{x}}_j$ being the approximation of the inputs computed by inverting the selected nonlinear principal components, as given by Eq. (21). The nonlinear nature of the input transformation does not allow direct inversion of the nonlinear model, and a separate model needs to be determined for the inversion transformation, ζ .

Several different techniques have been suggested for NLPCA. Kramer [65] uses an autoassociative neural network with three hidden layers. The first and second hidden layers determine the nonlinear input transformation, ϕ , and the second and third hidden layers determine the inverse mapping, ξ . Tan and Mavrouniotis [66] use an externally recurrent or input training neural network that manipulates the inputs to satisfy Eq. (15). Hastie and Stuetzle [45] and LeBlanc and Tibshirani [67] use statistical smoothers and splines, respectively, for determining the nonlinear projection. Dong and McAvoy [34] extend the method of Hastie and Stuetzle by using BPN instead of the statistical smoothers.

Nonlinear PLS. The approach for NLPLS based on linear projection of the inputs described in Section 3.1, may be extended to NLPLS based on nonlinear projection of the inputs. A BPN with three hidden layers, similar to that used for NLPCA [65] has been used for NLPLS [68]. The first and second hidden layers determine the nonlinear input transformation, and the second and third hidden layers determine the nonlinear basis functions. This network is trained by satisfying the following objective function

$$\min_{\alpha_m, \beta_m} \left\{ \sum_j \|\mathbf{x}_j - \hat{\mathbf{x}}_j\|^2 + \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \right\}.$$

This optimization criterion combines the objectives of finding the relationship between the inputs and minimizing the output prediction error, and is a compromise between the PPR and NLPCA objective functions. The performance of NLPLS with nonlinear projection deserves further study, but preliminary comparison with other empirical modeling methods indicates no clear advantages over neural network

methods in situations where the inputs are not nonlinearly related [68].

3.3. Partition-based methods

Partition-based methods include techniques such as classification and regression trees (CART) or recursive partitioning regression [69], inductive decision trees [70] and multivariate adaptive regression splines (MARS) [71]. The general empirical model given by Eq. (1) may be specialized to that determined by partition-based method as

$$\hat{y} = \sum_{m=1}^M \beta_m \theta_m \quad (X \in R_m), \quad (25)$$

where R_m is the set of the selected inputs. Restricting the inputs in the basis functions to a subset results in partitioning of the input space into hyper-rectangular regions, as illustrated in Fig. 1(d) and Fig. 2(f). This class of methods may also be considered to be a special case of methods based on linear projection with the projection hyperplane restricted to be perpendicular to at least one of the input axes, that is, projection directions can assume values of 0 or 1 only. These methods are included in a separate class due to their many unique characteristics. The basis functions used in partition-based methods may also be of a fixed or adaptive shape. Partition-based methods aim to derive a good set of regions in the input space and the corresponding basis functions that minimize the selected objective function. As in methods based on both linear and nonlinear projection, the regression coefficients in partition-based methods are usually determined to minimize the output prediction error. The explicit selection of the most relevant set of input variables results in the model determined by partition-based methods to be physically interpretable.

Classification and regression trees. The basis functions in a CART or inductive decision tree model are given by

$$\theta_m(X) = \prod_{p=1}^{P_m} H[s_{pm}(x_{v(p,m)} - t_{pm})], \quad (26)$$

where H is the Heaviside or step function

$$H[\eta] = 1 \quad \text{if } \eta \geq 0,$$

$$H[\eta] = 0 \quad \text{otherwise.}$$

P_m is the number of partitions or splits, $s_{pm} = \pm 1$ and indicates the right or left of the associated step function, $v(p, m)$ indicates the selected input variables in each partition, and t_{pm} represents the location of the split in the corresponding input space. The indices, p and m are used for the split and node or basis function, respectively. The basis functions given by Eq. (26) are of a fixed, piecewise constant shape.

The CART modeling method selects the variable for partitioning the input space as the one that minimizes the output mean-squares error of approximation. This recursive partitioning of the input space is continued until a large number of subregions or basis functions are generated. After splitting a region, it is removed from the model. Overfitting is avoided by penalizing the output prediction error for the addition of basis functions, and eliminating unnecessary splits by a backwards elimination procedure [69]. Thus, the objective function used for determining all the model parameters focuses entirely on minimizing the output error of approximation, and distribution of data in the input space or relationships among the input variables are not exploited. The CART model can be represented as a binary tree and is physically interpretable. But the discontinuous approximation at the partitions prohibits its application to model continuous input–output relationships. Furthermore, CART is often unable to identify interactive effects of multiple inputs.

Multivariate adaptive regression splines. MARS overcomes the disadvantages of CART for multivariate regression [71]. MARS divides the input space into overlapping partitions by keeping both the parent and daughter nodes after splitting a region. This prevents discontinuities in the approximation at the partition boundaries producing continuous approximations with continuous derivatives. As in CART, the MARS model is also represented by Eq. (25), but instead of using the fixed-shape and piecewise constant Heaviside or step function as the basis functions, MARS uses multivariate spline basis functions obtained by multiplying univariate splines represented by a two-sided truncated power basis,

$$\theta_m(X) = \prod_{p=1}^{P_m} H[s_{pm}(x_{v(p,m)} - t_{pm})]_+^q, \quad (27)$$

where q is the order of the splines. Comparison of

Eqs. (26) and (27) indicates that the value of $q=1$ results in the CART basis functions. For $q>0$, the approximation is continuous and has $q-1$ derivatives. The ability of MARS to provide significant insight into the input–output relationship is provided by rearranging Eq. (25) to

$$\hat{y} = \beta_0 + \sum_{P_m=1} y_i(x_i) + \sum_{P_m=2} y_{ij}(x_i, x_j) + \sum_{P_m=3} y_{ijk}(x_i, x_j, x_k) + \dots,$$

where each term indicates the input variables that are relevant to the model, and how they interact with other inputs in approximating the output. This representation greatly facilitates the physical interpretation of the MARS model. Empirical comparison of MARS with BPN has shown that the performance of MARS on problems with correlated inputs is often inferior to that of BPN [32]. This observation may be explained by the fact that for approximating correlated inputs efficiently, the projection directions should be able to assume any value, but MARS restricts its projection directions to assume the value of 0 or 1 only to improve the physical interpretability of the model.

4. Illustrative example

The insight into empirical modeling methods provided by the common comparison framework developed in this paper is further enhanced by the illustrative example presented in this section. The purpose of this example is not to compare the performance of various methods, or to recommend any method, but to provide insight and a deeper understanding of the similarities and differences between various techniques. Empirical comparison of various methods may be found in other papers [35,36,72].

This example consists of two inputs and one output related as

$$y = (x_1 + 0 \times x_2)^2 = x_1^2. \quad (28)$$

The input weights in Eq. (28) indicate that the optimum projection direction for this surface is $\alpha = [1 \ 0]$, which is the direction parallel to the x_1 -axis. The training set consists of 189 data points selected from the quadratic surface shown in Fig. 4(a) to form a narrow band, as depicted in Fig. 4(b). The error of

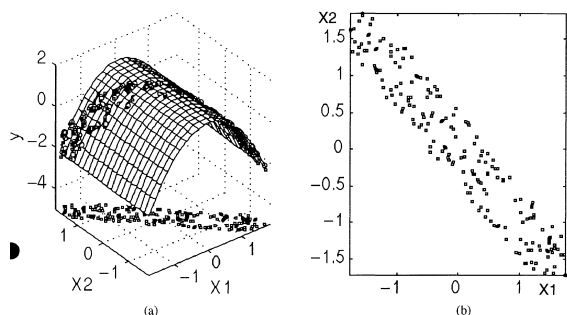


Fig. 4. Nature of hypersurface and available training data for illustrative example.

Table 2

Performance of various empirical modeling methods on parabola example

Method	Training MSE	Testing MSE
OLS	9.94E-1	7.49E-1
PCR	9.96E-1	7.76E-1
PLS	9.94E-1	7.49E-1
BPN	4.88E-6	4.73E-6
PPR	2.08E-6	2.05E-6
NLPCR	8.98E-2	7.68E-2
NLPLS	2.33E-2	2.32E-2
RBFN	4.90E-2	4.90E-2
MARS	9.21E-4	8.39E-4

approximation based on training and testing data for each method is summarized in Table 2. The testing data consist of 95 data points.

OLS finds the projection directions that maximize the correlation between the projected training data and the outputs. The training data projected on the OLS plane and the linear basis function are shown in Fig. 5(a), and the projection direction in Fig. 5(b). PCR finds the projection directions that capture the maximum variance of the projected inputs. Consequently, the first principal component captures the linear relationship between the inputs and the second component is perpendicular to it, as shown in Fig. 6(b) and (d). The projected training data and linear basis function that minimizes the mean-squares error in the latent variable-output space are as shown in Fig. 6(a) and (c). The projection directions determined by PLS are depicted in Fig. 7(b) and (d), and lie between those determined by OLS and PCR. Due to the quadratic nature of the hypersurface, all of these linear regres-

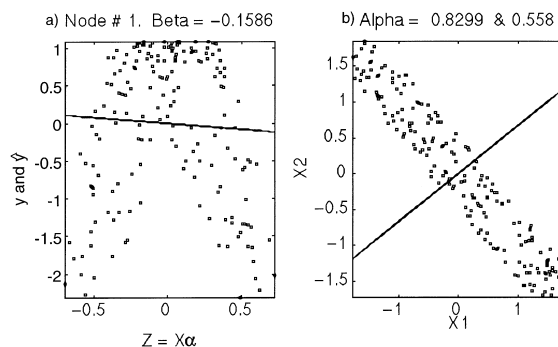


Fig. 5. Results of OLS modeling: (a) data in projected input-output space (dots) and linear activation function (line); (b) data in input space (dots) and projection direction (line) that minimizes the output mean-squares error.

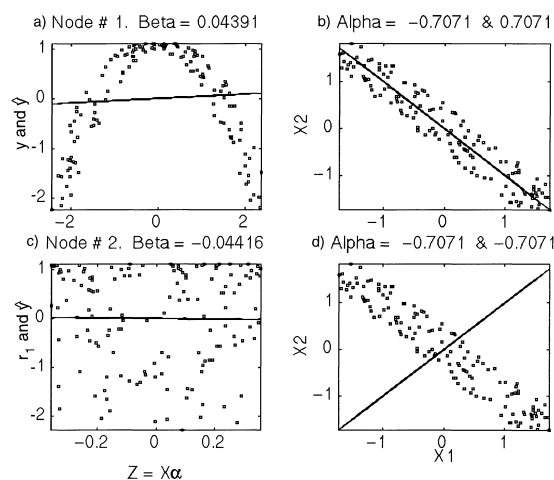


Fig. 6. Results of PCR modeling: (a) data in projected input-output space (dots) and linear activation function (line) for first node; (b) data in input space (dots) and projection direction (line) for first node that maximizes the captured variance; (c) data representing residual error first node; and activation function for second node; (d) data and projection direction for second node that maximizes the captured residual input variance.

sion methods result in a relatively large error of approximation, as shown in Table 2.

Nonlinear modeling methods are much better for approximating the quadratic surface given by Eq. (28). A BPN with two hidden nodes is able to combine the two sigmoid basis functions to approximate the quadratic surface, as shown in Fig. 8(a). PPR is also able to approximate the quadratic surface with only one basis function as shown in Fig. 8(b). Both BPN and PPR find the actual projection direc-

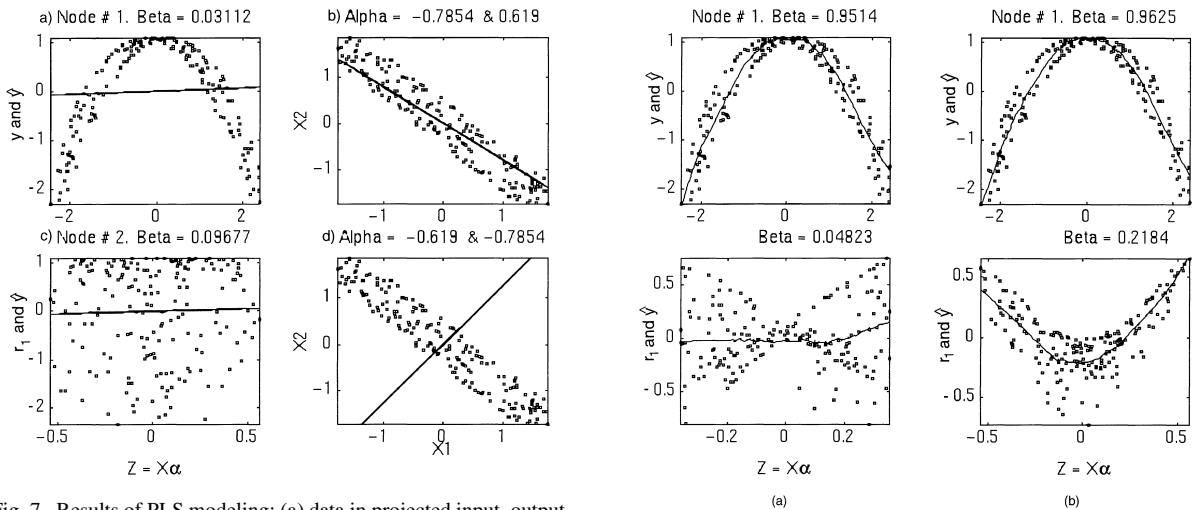


Fig. 7. Results of PLS modeling: (a) data in projected input–output space (dots) and linear activation function (line) for first node; (b) data in input space (dots) and projection direction (line) for first node that maximizes the covariance between projected input and output; (c) data and activation function for second node based on residual error; (d) data and projection direction of second node.

tions as shown in Fig. 8(c). Comparison of the performance of BPN and PPR indicates that PPR possesses several advantages over BPN [53]. Due to the node-by-node training methodology and the adaptive basis functions, PPR finds the best possible model and projection directions for any number of nodes, whereas BPN finds the correct projection directions

Fig. 9. (a) Activation functions for NLPCR. Projection directions are the same as that for linear PCR, as shown in Fig. 7; (b) activation functions for NLPLS with projection directions shown in Fig. 8.

only when the optimum number of nodes are selected for the hidden layer. The model determined by PPR also requires less basis functions than that determined by BPN.

The performance of NLPCR and NLPLS is depicted in Fig. 9(a) and (b), and is better than their linear versions, but unlike PPR and BPN, neither method is

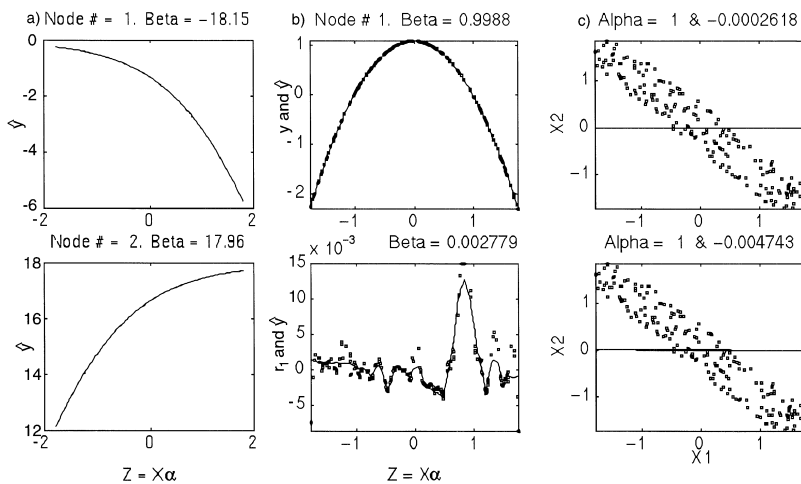


Fig. 8. Result of BPN and PPR modeling with two nodes: (a) BPN sigmoid activation functions; (b) PPR activation functions of adaptive shape; (c) projection directions for BPN and PPR that minimize the output error of approximation.

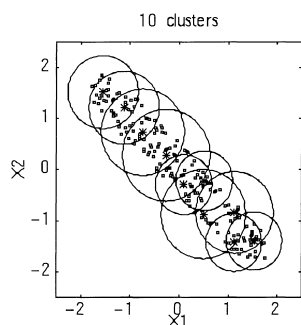


Fig. 10. Clusters determined by nearest neighbor clustering for modeling by RBFN.

successful in determining the correct projection direction of [10]. The performance of NLPCR and NLPLS is expected to be better than that of PPR and BPN when the ratio of training data to inputs is small, and dimensionality reduction becomes more important [31,33,35].

The clusters determined by nearest-neighbor clustering are shown in Fig. 10. These clusters form the basis functions for an RBFN model, whose error is listed in Table 2. The number of clusters is determined via nearest neighbor clustering and crossvalidation. The resulting error of approximation on testing data is several orders of magnitude larger than that for BPN or PPR, since RBFNs like PCR do not find the best input transformation for minimizing the output error of approximation. Consequently, like PCR, RBFNs are best for problems where the data are sparsely distributed in the input space, and dimensionality reduction by nonlinear projection on localized clusters, is important. The performance of MARS is depicted in Fig. 11 with five knots. MARS is able to clearly identify x_1 as

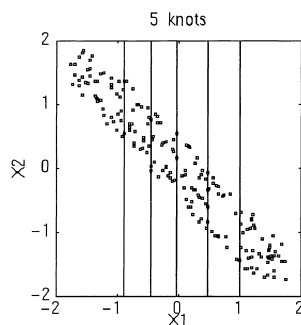


Fig. 11. Relevant variable, x_1 , and knot locations determined by MARS.

the important variable for this problem, since placing knots on variable x_2 is not necessary.

5. Challenges for unifying empirical modeling methods

The insight provided by the common framework described in Section 2 is useful for understanding the features of linear and nonlinear empirical modeling methods, and is expected to facilitate the development of hybrid methods possessing properties desired by the user. The three criteria identified for comparing various methods indicate that combining the properties of existing empirical modeling methods requires development of the following techniques:

- *Input transformation methods* that can specialize to the best linear or nonlinear projection, or input partition, depending on the demands of the modeling task.
- *Basis functions* that can assume any shape, and specialize to linear or nonlinear functions as required by the user or by the nature of the problem.
- *Optimization criteria* that subsume the criteria used by various empirical modeling methods, and can specialize to the appropriate criterion depending on the modeling task.
- *Training methodology* that is efficient and uses the general input transformation, basis functions, and optimization criteria for empirical modeling.

Such unification of empirical modeling methods is expected to be helpful in selecting the best empirical modeling method for a given task, and in developing techniques that combine the desired properties of various existing approaches. A novel technique that unifies all methods based on linear projection has been developed based on the framework presented in this paper, and is described by Bakshi and Utojo [35]. This method, called nonlinear continuum regression, can specialize to any method on the continuum between methods based on linear projection, depending on the nature of the problem.

6. Conclusions

A common framework for comparing various empirical modeling methods and bringing out their

similarities and differences has been presented. This framework is based on the representation of empirical models as expansion on a set of basis functions, and the realization that various empirical modeling methods may be obtained from the general model depending on decisions about only three criteria: the nature of the input transformation, type of activation functions, and optimization criteria for estimating the model parameters. According to the nature of the input transformation, empirical modeling methods may be categorized into methods based on linear projection, methods based on nonlinear projection, and partition-based methods. The activation functions in each method may be of fixed or adaptive shape, and the optimization criteria for estimating the model parameters determine the generality or bias of the resulting model. This common framework is used to provide an overview of the wide variety of empirical modeling methods popular in chemometrics and chemical engineering, and permits the development of novel empirical modeling methods that combine various existing techniques.

7. Nomenclature

b_j	OLS and RR coefficient for j th input
\mathbf{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
\mathbf{r}_m	output residual vector approximated by m th node
\mathbf{t}_m	cluster center or knot location vector
\hat{x}	approximated input
x_{ij}	element of \mathbf{X}
\mathbf{x}_j	j th column of \mathbf{X}
\mathbf{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, $I \times K$
\mathbf{z}_m	m th latent variables 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
λ	ridge regression parameter
θ_m	m th node or basis function
$\ \cdot\ $	sum of the squares of each element in the argument vector

Acknowledgements

Partial financial support from an Ohio State University Seed Grant is gratefully acknowledged.

References

- [1] R. Schindler, R. Vonach, M. Watkins, *Anal. Chem.* 70 (1998) 226.
- [2] J.V. Kresta, J.F. MacGregor, T.E. Marlin, *Can. J. Chem. Eng.* 69 (1991) 35.
- [3] R.D. De Veaux, L.H. Ungar, J.M. Vison, *Proc. Am. Cont. Conf.*, Baltimore, MA, 1994.
- [4] B.M. Wise, N.L. Ricker, *Proc. Cont. Qual.* 4 (1992) 77.
- [5] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [6] B.S. Dayal, J.F. MacGregor, *J. Proc. Cont.* 7 (1997) 169.
- [7] H. Martens, T. Naes, *Multivariate Calibration*, Wiley, New York, 1989.
- [8] J.E. Jackson, *A User's Guide to Principal Components*, Wiley, New York, 1991.
- [9] D. Svozil, V. Kvasnicka, J. Pospichal, *Chemom. Intell. Lab. Sys.* 39 (1997) 43.
- [10] S. Nakai, E. Li-Chan, *Crit. Rev. Food Sci. Nutr.* 33 (1993) 477.
- [11] J.A. Leonard, M.A. Kramer, *IEEE Control Systems*, (1991) 31.
- [12] N.V. Bhat, P.A. Minderman, T.J. McAvoy, N.S. Wang, *IEEE Cont. Sys. Mag.* (1990) 24.
- [13] L.H. Ungar, B.A. Powell, S.N. Kamens, *Comp. Chem. Eng.* 14 (1990) 561.
- [14] G. Katerman, *Chemom. Intell. Lab. Sys.* 19 (1993) 135.
- [15] S. Sekulic, M.B. Seasholtz, Z. Wang, B.R. Kowalski, S.E. Lee, B.R. Holt, *Anal. Chem.* 65 (1993) 835.
- [16] S. Wold, *Chemom. Intell. Lab. Sys.* 14 (1992) 71.
- [17] B.R. Bakshi, G. Stephanopoulos, *Comp. Chem. Eng.* 18 (1994) 303.
- [18] B. Joseph, F.H. Wang, D.S.-S. Shieh, *Comp. Chem. Eng.* 16 (1992) 413.
- [19] S.J. Qin, T.J. McAvoy, *Comp. Chem. Eng.* 20 (1996) 147.
- [20] G. Cybenko, *Continuous valued neural networks with two hidden layers are sufficient*. Technical Report, Department of Computer Science, Tufts University, 1988.

- [21] T. Poggio, F. Girosi, A theory of networks for approximation and learning, A.I. Memo 1140, MIT, MA, 1989.
- [22] T. Poggio, F. Girosi, B.F. Hyper, A powerful approximation technique for learning, in: P.H. Winston, S.A. Shellard (Eds.), *Artificial Intelligence at MIT*, MIT Press, Cambridge, 1990.
- [23] A. Lorber, L.E. Wangen, B.R. Kowalski, *J. Chemom.* 1 (1987) 19.
- [24] M. Stone, R.J. Brooks, *J. Roy. Stat. Soc., Ser. B.* 52 (1990) 237.
- [25] I.E. Frank, J.H. Friedman, *Technometrics* 35 (1993) 109.
- [26] P. Diaconis, M. Shahshahani, *SIAM J. Sci. Stat. Comput.* 5 (1984) 175.
- [27] P.J. Huber, *Ann. Stat.* 13 (1985) 58.
- [28] D.L. Donoho, I.M. Johnstone, *Ann. Stat.* 17 (1989) 58.
- [29] M.J. Piovoso, A.J. Owens, in: Y. Arku, W.H. Ray (Eds.), *Chemical Process Control CPC IV*, CACHE, Austin, TX, 1986.
- [30] S. Bannour, M.R. Azimi-Sadjadi, *IEEE Trans. Neur. Net.* 6 (1995) 457.
- [31] T.R. Holcomb, M. Morari, *Comp. Chem. Eng.* 16 (1992) 393.
- [32] R. DeVeaux, D. Psychogios, L.H. Ungar, *Comp. Chem. Eng.* 17 (1993) 819.
- [33] S.J. Qin, T.J. McAvoy, *Comp. Chem. Eng.* 16 (1992) 379.
- [34] D. Dong, T.J. McAvoy, *Comp. Chem. Eng.* 20 (1996) 65.
- [35] B.R. Bakshi, U. Utojo, *Comp. Chem. Eng.* 1998, 22 (1998) 1859.
- [36] I.E. Frank, *Chemom. Intell. Lab. Sys.* 27 (1995) 1.
- [37] B. Cheng, D.M. Tetterington, *Stat. Sci.* 1 (1994) 2.
- [38] B.D. Ripley, *J. Roy. Stat. Soc.* 56 (1994) 409.
- [39] W.S. Sarle, Neural networks and statistical models, Proceedings of the Nineteenth Annual SAS Users Group International Conference, 1994.
- [40] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P. Glorennec, H. Hjalmarsson, A. Juditsky, *Automatica* 31 (1995) 1691.
- [41] J.R. Rice, *The Approximation of Functions*, vols. I, II, Addison-Wesley, Reading, MA, 1964.
- [42] B.R. Bakshi, G. Stephanopoulos, *AIChE J.* 39 (1993) 57.
- [43] N.R. Draper, H. Smith, *Applied Regression Analysis*, Wiley, New York, 1981.
- [44] P. Geladi, B.R. Kowalski, *Anal. Chim. Acta.* 185 (1986) 1.
- [45] T.J. Hastie, W. Stuetzle, *J. Am. Stat. Assoc.* 84 (1989) 505.
- [46] S. Wold, in: K. Joreskog, H. Wold (Eds.), *Systems Under Indirect Observation*, Elsevier, Amsterdam, 1982.
- [47] A. Hoskuldsson, *J. Chemom.* 2 (1988) 211.
- [48] D.E. Rumelhart, J.L. McClelland et al., *Parallel Distributed Processing*, vol. 1, MIT Press, Cambridge, MA, 1986.
- [49] S.E. Fahlman, C. Lebiere, *Advances in neural information processing systems*, Morgan Kaufmann 2 (1990) 524.
- [50] J.H. Friedman, W. Stuetzle, *J. Am. Stat. Assoc.* 76 (1981) 817.
- [51] J.H. Friedman, A. variable span smoother, Technical Report 5, Department of Statistics, Stanford University, 1984.
- [52] C.B. Roosen, T.J. Hastie, *J. Comput. Graph. Stat.* 3 (1994) 235.
- [53] J.N. Hwang, M. Lay, R.D. Martin, J. Schimert, *IEEE Trans. Neur. Net.* 5 (1994) 342.
- [54] N.B. Vogt, *Chemom. Intell. Lab. Sys.* 7 (1989) 119.
- [55] I.E. Frank, Presented at InCINC'94, the First International Chemometrics Internet Conference, 1994.
- [56] I.E. Frank, *Chemom. Intell. Lab. Sys.* 8 (1990) 109.
- [57] S. Wold, N. Kettaneh-Wold, B. Skagerberg, *Chemom. Intell. Lab. Sys.* 7 (1989) 53.
- [58] D. Wienke, L. Buydens, *Trends Anal. Chem.* 14 (1995) 398.
- [59] G.A. Carpenter, S. Grossberg, *Appl. Optics* 26 (1987) 4919.
- [60] G.A. Carpenter, S. Grossberg, J.H. Reynolds, *Neur. Net.* 4 (1991) 565.
- [61] S. Chen, C.F.N. Cowan, P.M. Grant, *IEEE Trans. Neur. Net.* 2 (1991) 302.
- [62] J. Moody, *Fast Learning in Multi-Resolution Hierarchies*, Research Report, Yale University, YALEU/DCS/RR-681, 1989.
- [63] S.N. Kavuri, V. Venkatasubramanian, *Comp. Chem. Eng.* 17 (1993) 765.
- [64] R.P. Lippmann, *IEEE ASSP Mag.* (1987) 4.
- [65] M.A. Kramer, *AIChE J.* 37 (1991).
- [66] S. Tan, M.L. Mavrovouniotis, *AIChE J.* 41 (1995) 1471.
- [67] M. LeBlanc, R. Tibshirani, *J. Am. Stat. Assoc.* 89 (1994) 53.
- [68] E.C. Malthouse, R.S.H. Mah, A.C. Tamhane, presented at InCINC'94, the First International Chemometrics Internet Conference, 1994.
- [69] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees*, Wadsworth, California, 1984.
- [70] J.R. Quinlan, *Induction of decision trees*, *Machine Learning* 1 (1986) 81.
- [71] J.H. Friedman, *Ann. Stat.* 19 (1991) 1.
- [72] B.M. Wise, B.R. Holt, N.B. Gallagher, S. Lee, Presented at InCINC'94, the First International Chemometrics Internet Conference, 1994.