Journal of Alloys and Compounds 279 (1998) 39–46

Unification of neural and statistical methods as applied to materials structure-property mapping

Bhavik R. Bakshi*, Raja Chatterjee

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

Abstract

A wide variety of neural and statistical methods are available for nonlinear empirical modeling based on different modeling approaches. Selecting the best method for a given task requires deep understanding of their similarities and differences and a systematic approach to method selection. This paper presents a common framework for gaining insight into neural and statistical modeling methods. The framework is then used to unify methods that combine inputs by linear projection before applying the basis function. The result of this unification is a new method called nonlinear continuum regression (NLCR) that unifies 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 partial least squares regression (NLPLS), and nonlinear principal component regression (NLPCR), by spanning the continuum between these methods. The unification is facilitated by developing a common objective function for all methods in this category, and an efficient hierarchical training algorithm, illustrative examples on synthetic data and materials structure-property prediction demonstrate the ability of NLCR to specialize to the best existing method based on linear projection, or to a method between existing methods, resulting in the most general model from this class of methods. © 1998 Elsevier Science S.A. All rights reserved.

Keywords: Statistics; Neural networks; Materials processing; Empirical modeling

1. Introduction

In addition to artificial neural networks (ANN), several statistical methods are also available for nonlinear empirical modeling. These methods include, projection pursuit regression (PPR), nonlinear principal component regression (NLPCR), nonlinear partial least squares regression (NLPLS), classification and regression trees (CART), and multivariate adaptive regression splines (MARS). These neural and statistical empirical modeling methods differ in their modeling approach, causing some methods to perform better for certain types of modeling problems. For example, backpropagation networks (BPN) often require a large amount of training data to obtain an acceptable model for a given number of input variables, whereas, statistical methods such as, NLPCR and NLPLS can perform equally well with a smaller ratio of training data to input variables. ANN usually provide black box models, whereas the model obtained by CART or MARS may be represented in terms of simple rules. Statistical methods with adaptive basis functions such as PPR, usually require less basis functions for comparable performance than neural techniques such as, BPN.

Given this broad variety of empirical modeling methods, it is important to select the best method for a given modeling task. Proper selection requires a deep understanding of all the modeling methods and a systematic approach to model selection, neither of which are easily available. This paper presents a common framework for comparing empirical modeling methods and enabling greater understanding of their similarities and differences. This framework is based on representing the model developed by any empirical modeling method as a weighted sum of basis functions, and showing how various methods can be derived depending on decisions about the nature of the input transformation, the type of activation or basis functions, and the optimization criteria for determining the adjustable parameters. The insight provided by this comparison framework is then used to unify linear and nonlinear empirical modeling methods that combine the inputs as a linear weighted sum before operation of the basis function. These methods based on linear projection include linear methods such as, ordinary least squares regression (OLS), partial least squares regression (PLS),

*Corresponding author.

0925-8388/98/$19.00 © 1998 Elsevier Science S.A. All rights reserved.
PII: S0925-8388(98)00610-0

principal components regression (PCR) and ridge regression (RR), and nonlinear methods such as, backpropagation networks with a single hidden layer, projection pursuit regression, nonlinear partial least squares regression, and nonlinear principal component regression. The comparison framework shows that all methods based on linear projection are special cases along a continuum of methods. The result of their unification is a new method called Non-Linear Continuum Regression (NLCR) that can specialize to any existing method or to a method along the continuum between existing methods, with the help of an additional tuning parameter. An efficient hierarchical training methodology is developed for NLCR modeling that trains one node at a time to reduce the residual error of approximation. 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.

2. A common comparison framework for empirical modeling methods

The model determined by any empirical modeling method may be represented as a weighted sum of basis functions,

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

where, $\hat{y}_k$ is the k-th predicted output or response variable, θ_m is the m-th basis or activation function, β_{mk} is the output weight or regression coefficient relating the m-th basis function to the k-th output, α is the matrix of basis function parameters, ϕ_m represents the input transformation, and x_1, ..., x_J are the inputs or predictor variables. The variable obtained by transforming the inputs, $\mathbf{z}_m = \phi_m(\alpha;\mathbf{x})$, is often referred to as the latent variable or projected input. Specific empirical modeling methods may be derived from Eq. (1) 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 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

Reducing the dimensionality of the input space is essential for improving the complexity of the modeling task, and the quality of the extracted model. Empirical modeling techniques fight this 'curse of dimensionality' by transforming the inputs to latent variables that capture the relation between the inputs 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 the following 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, as shown in Fig. 1(a), before applying the basis function. Thus, the inputs are transformed by combination as a linear weighted sum to form the latent variables.
- 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, as shown in Fig. 1(b) and (c). If the inputs are projected on a localized hypersurface then the basis functions are local, as depicted in Fig. 1(c). 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, as depicted in Fig. 1(d).

2.2. Type of activation or basis functions

The wide variety of activation or basis functions used in empirical modeling methods may be broadly divided into the following two categories:

- Fixed-shape basis functions. The basis functions in several empirical modeling methods are of a fixed

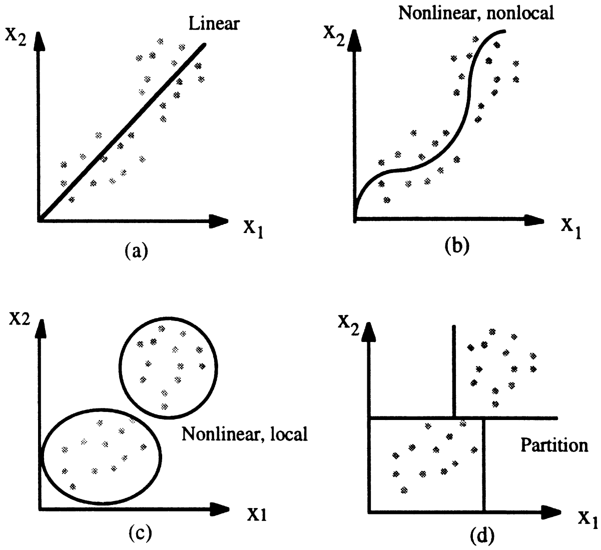

Fig. 1. Input transformation in (a) methods based on linear projection, (b), and (c) methods based on nonlinear projection, non-local and local transformation respectively, and (d) partition-based methods.

shape such as, linear, sigmoid, Gaussian, wavelet, or sinusoid. Adjusting the basis function parameters changes their location, size, and orientation, but their shape is decided a priori, and remains fixed.

- Adaptive-shape basis functions. 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.

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, *f*, and parameters, *a*, whereas the model relating the transformed inputs to the output is determined by the parameters, *b*, and basis functions, *q*. Empirical modeling methods often use different objective functions for determining the input transformation, and the model relating the transformed inputs to the output. This separation of the empirical modeling optimization criteria provides explicit control over the dimensionality reduction by input transformation, and often results in more general empirical models. Most empirical modeling methods minimize the mean square error of approximation to determine the basis function, *q* and regression coefficients, *b*.

The criterion used to determine the input transformation parameters, ϕ and α differ for each method depending on the emphasis on transforming the inputs versus minimizing the output error of approximation. For example, PCR and NLPCR focus entirely on obtaining an optimum transformation of the inputs by maximizing the variance captured by the latent variables, whereas, OLS, BPN, and PPR transform the inputs to minimize the output prediction error, and PLS and NLPLS maximize the covariance between the projected inputs and output. 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 unifying methods based on linear projection, as described in the next section.

3. Unification of methods based on linear projection

The latent variable for methods based on linear projection is a weighted sum of the inputs. The resulting model may be represented by specializing Eq. (1) to,

$$\hat{y}_k = \sum_{m=1}^{M} \beta_{mk} \theta_m \left(\sum_{j=1}^{J} \alpha_{jm} x_j \right) \tag{2}$$

.

The comparison framework described in the previous section indicates that unification of methods based on linear projection requires common methods for determining the different shapes of basis functions, a common objective function and a common training methodology. Such a unified method is developed in this section for modeling with multiple inputs and a single output.

3.1. Techniques for determining basis functions

Each basis function for methods based on linear projection maps the linearly projected input, $\mathbf{z}_m$ to the output, $\mathbf{y}$.

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 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 proj., nonlocal	Fixed shape, sigmoid	$[\alpha, \beta]$ – min. output prediction error
NLPCA	Nonlinear proj., nonlocal	Adaptive shape	$[\alpha, \phi]$ – min. input 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

Unification of the variety of basis functions used in these methods requires a general approach that can provide any linear or nonlinear relationship between the latent variable and output, depending on the nature of the training data. Such basis functions may be obtained by using univariate smoothing techniques for approximating the training data in the projected input–output space. A variety of such smoothing techniques are available including, variable span smoothers [6], Hermite functions [8], automatic smoothing splines [10], and backpropagation networks. The NLCR method developed in this paper can use any of these smoothing techniques to determine the appropriate basis functions.

3.2. General optimization criterion for projection directions

Unification of methods based on linear projection 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. Thus, the optimization criterion should span the continuum between different methods based on linear projection. The techniques of PCR and NLPCR lie at one extreme of this continuum, since their optimization criterion is unaffected by the nature of the outputs or basis functions. Both methods focus on transforming only the input space by maximizing the variance captured by the projected inputs as,

$$\max_{\alpha_m} \left\{\mathrm{var}(\mathbf{X}\alpha_m)\right\} \tag{3}$$

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 [1] and may be written as,

$$\max_{\alpha_m} \left\{\mathrm{corr}^2(\mathbf{y},\theta_m(\mathbf{X}\alpha_m))\right\} \tag{4}$$

The optimization criteria at two extremes of the continuum of methods given by Eq. (3) and Eq. (4) may be combined as,

$$\max_{\alpha_m} \left\{\mathrm{corr}^2(\mathbf{y},\theta_m(\mathbf{X}\alpha_m))\mathrm{var}(\mathbf{X}\alpha_m)\right\} \tag{5}$$

and should result in a method between PPR and NLPCR. Indeed, Eq. (5) has been used as the optimization criterion for NLPLS modeling by [13] for quadratic PLS, [14] for spline PLS, and [7] for neural net/PLS.

Eq. (3), Eq. (4) and Eq. (5) may be combined to obtain a general optimization criterion that subsumes all methods based on linear projection as,

$$\max_{\alpha_m} \left\{[\mathrm{corr}^2(\mathbf{y},\theta_m(\mathbf{X}\alpha_m))][\mathrm{var}(\mathbf{X}\alpha_m)]^{\gamma}\right\} \tag{6}$$

where values of γ equal to 0, 1, and ∞ result in BPN, PPR or OLS; NLPLS or PLS; and NLPCR or PCR, respectively. Eq. (6) is a nonlinear version of the optimization criterion suggested by [12] to unify OLS, PLS, and PCR. The exponents in Eq. (6) may be modified to,

$$\max_{\alpha_m} \left\{[\mathrm{corr}^2(\mathbf{y},\theta_m(\mathbf{X}\alpha_m))]^{1+\gamma-2\gamma^2}[\mathrm{var}(\mathbf{X}\alpha_m)]^{3\gamma-2\gamma^2}\right\} \tag{7}$$

This objective function reduces to various existing methods, as summarized in Table 2. The effect of the adjustable parameter, y, on the generality of the empirical model may be understood in terms of the bias-variance trade-off. 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 NLCR training methodology aims to find this value of y that optimizes the bias-variance trade-off as described in the next section.

The remaining adjustable parameters, namely the regression coefficients, β_m and basis functions, θ_m are determined by minimizing the mean-squares error of approximation,

$$\min_{\beta_m,\theta_m} \frac{1}{I}\sum_{i=1}^{I}(y_i-\hat{y}_i)^2 \tag{8}$$

Eq. (7) and Eq. (8) constitute the general objective function that unifies all methods based on linear projection.

3.3. 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, and the common optimization criterion, to determine the empirical model in an efficient manner. Training methodologies for empirical model building may determine the model parameters simultaneously for all the basis functions, or hierarchically for one basis function at a time. Examples of the simultaneous approach include eigenvalue decomposition for computing the projection directions in PCR and PLS, and the error backpropagation algorithm for BPN [11]. Examples of the hierarchical approach include the Nonlinear Iterative Partial Least Squares (NIPALS) algorithm [9] for PCR and PLS, cascade correlation for BPN [4], and the PPR algorithm [5]. Hierarchical modeling methods are usually

Table 2
Specialization of objective function 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

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.

The steps comprising the hierarchical, node-by-node NLCR training methodology, are shown below.

1. For $\gamma \leftarrow 1$ to 0,
2. Add new node and optimize,
3. Projection directions,
4. Basis functions, θ_m
5. Regression coefficients, β_m
6. Update model
7. Update output residual
8. Update input residuals or backfit previously added nodes
9. If prediction error is acceptable, go to ten, else go to two

10.End

The projection directions are computed by optimizing the general objective function for the selected value of γ, for the basis function and regression coefficient determined in the previous iteration. If orthonormal projection directions are desired, as in PCR and PLS, then both the input and output residuals need to be updated, otherwise, the input residual is left unchanged. The modeling ability of each node may be improved in Step (8) by accounting for the nature of previously added basis functions by adjusting their parameters by backfitting or backward pruning [6].

The NLCR training methodology can specialize 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 determining the input and output residuals after training each node. Backfitting is not needed since the projection directions are fixed by the orthogonality requirement. 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.

Efficient techniques for finding the best value of γ are essential for the application of NLCR modeling to practical problems. The optimum value of γ may be found from models developed for several values, and selecting the y 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 trial-and-error approach computationally expensive for large problems. Furthermore, the modeling with several different initial values of the parameters may be necessary 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, which is the eigenvector of the input covariance matrix.
- Decreasing the value of y causes the projection directions to gradually rotate away from those capturing 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 is currently being explored, and detailed results will be included in the final version of this paper. Preliminary examples show that it provides reproducible results and decreases the computation time for NLCR modeling.

4. Illustrative example

The properties of NLCR are illustrated by solving the following examples based on synthetic data. The NLCR training methodology is implemented in Matlab, and is available from the corresponding author.

4.1. Parabola example

This set of illustrative examples is based on data generated by the following model,$y = t_1^2$ $x_1 = t_1 + 0.1\varepsilon_1$, $x_2 = t_2 + 0.1\varepsilon_2$, $x_3 = \sigma_3\varepsilon_3$, $x_4 = \sigma_4\varepsilon_4$where, ε_i, $i =$ 1, ..., 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 as shown in Fig. 2. This model indicates that the optimum projection direction for this problem is approximately [1 0 0 0]. The performance of NLCR is compared for 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 [2].

The results of NLCR modeling with 5 training data with $\sigma_3 = \sigma_4 = 1$ are summarized in Table 3. As expected, the

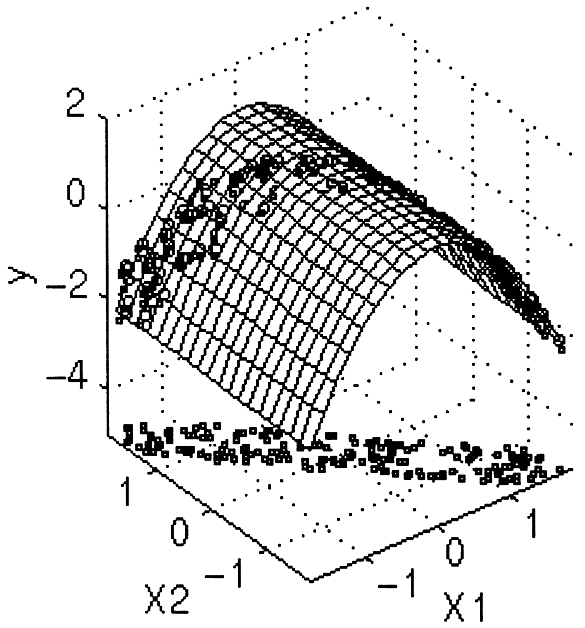

Fig. 2. Data for parabola example without noise.

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 error of 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 the last column of Table 3, the performance of PPR/BPN is several orders of magnitude worse than that of other methods based on linear projection.

The results of NLCR modeling for different number of training data after decreasing the value of σ_3 and σ_4 to 0.1 indicate that as the number of training data increases, the value of γ for the best NLCR model shifts towards zero. For five training data the model with the smallest error of approximation for testing data has $\gamma=0.15$, for ten training data, $\gamma=0.1$, and for fifty training data, $\gamma=0.02$. This behavior of γ indicates that obtaining a biased model by capturing the relationship between the inputs is less important when the ratio of training data to input variables is large. The optimum projection directions also tend to be closer to [1 0 0 0] as the amount of training data increases, and the basis function starts looking more like the underlying parabola.

Table 4
Data from Table 1, cluster 2 of [16], as used for NLCR modeling

Gap	*a*	*c*	Radius ion	Density	At. wt.
2.1	4.61	4.61	29	0.0001	84.6
2.26	4.359	4.359	29	3.191	40.09
3.3	3.251	5.209	22	5.651	81.369
3.9	3.823	6.261	53	3.536	97.434
4	5.481	5.171	22	4.502	181.836
5.9	5.58	4.69	22	0.0001	60.069
6	4.359	4.359	29	3.191	40.09
6.2	3.11	4.98	25	3.255	40.99
7	7.45	6.97	59	0.0001	136.086
8.4	4.9134	5.4052	22	2.65	60.078
4	5.481	5.171	22	4.502	181.836

4.2. Materials structure-property prediction

This case study models the relationship between various material properties with the objective of predicting the properties of new materials. The training and testing data for all the case studies are identical to those used by three other papers in this special issue: Jackson, et al. using rough sets [16], Chen et al. [15], using the OFBNN and Pao and Meng using the ratio-constrained mapping [17].

In Table 4, Table 5 and Table 6 the data used in this case study are shown. In each of the three tables, the last column (atomic weight) contains the dependent variable. The first five columns contain the input variables. The last row in each table, contains the testing set, while the remaining rows make up the training set. The data in the three tables have been broken down into subgroups from a larger data base according to clusters determined by [15].

For the data in Table 4, the best NLCR model was obtained for $\gamma=0$, with eight basis functions determined by the supersmoother [6] (Fig. 3). The training MSE is 0.00085 based on original data. The desired feature value is 181.8360 and the predicted value is 181.8357. The loading directions, α, were initialized using results from principal component regression with $\gamma=1$. Models for

Table 3
Results of NLCR modeling with five training data, 95 testing data, $\sigma_3=\sigma_4=1$

γ	a_{11}	a_{21}	a_{31}	a_{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
0.85	0.6909	−0.7014	0.0895	−0.1506	0.8609	5.887e-02	5.318e-01
0.65	0.6873	−0.7085	0.1259	−0.0985	0.8667	4.883e-02	5.822e-01
0.45	0.6793	−0.7113	0.1706	−0.0591	0.8716	4.032e-02	7.295e-01
0.25	0.6607	−0.7088	0.2469	−0.0121	0.8783	2.866e-02	1.196e+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

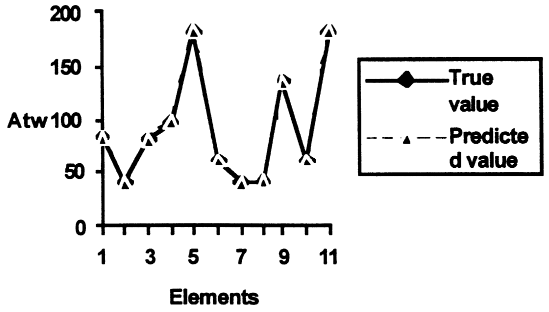

Fig. 3. Result of NLCR model for data in Table 4.

Table 5
Data from Table 2, cluster 1 of [15], as used for NLCR modeling

Gap	*a*	*c*	Radius ion	Density	At. wt.
0.57	5.943	11.217	71	5.6	334.97
1.2	6.099	11.691	66	5.808	286.798
1.53	5.489	11.101	53	4.73	242.468
1.7	5.606	11.006	66	4.73	242.468
1.74	5.606	10.88	71	4.7	243.43
1.8	5.981	10.865	66	5.759	335.51
2.1	4.145	9.496	53	7.101	232.654
2.43	5.351	10.47	53	4.332	197.388
2.638	5.751	10.238	53	4.66	241.718
2.91	5.74	10.776	59	4.549	246.93
3.05	5.568	10.04	53	3.97	380.096
2.05	5.463	10.731	59	4.105	199.9

Table 6
Data from Table 2, cluster 3 of [15], used for NLCR modeling

Gap	*a*	*c*	Radius ion	Density	At. wt.
0.23	6.479	6.479	89	5.777	236.55
0.33	4.457	5.939	82	6.25	236.55
0.36	6.268	6.479	71	5.72	189.79
0.72	6.095	6.095	89	5.615	191.47
1.35	5.868	5.868	59	4.798	145.77
1.4	5.653	5.653	71	5.316	144.71
1.7	4.361	4.954	66	4.819	78.96
2.3	6.101	6.101	82	5.924	192.97
2.7	5.667	5.668	66	5.318	144.33
2.8	6.473	6.473	126	6.0	234.77
2.91	5.69	5.69	82	4.72	143.449
2.95	6.042	6.042	96	5.667	190.44
3.05	5.568	5.568	53	3.97	80.096
3.17	5.405	5.405	77	4.137	98.993
2.3	5.45	5.45	59	4.135	100.69

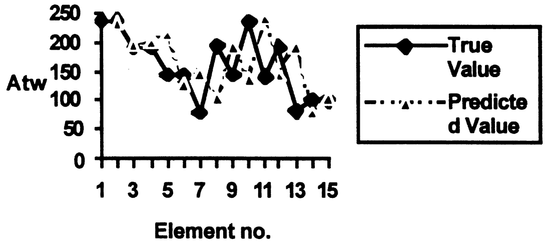

Fig. 5. Result of NLCR model for data in Table 6.

smaller values of γ were developed with the initial parameters determined by the previous larger value of γ. Such an initialization of the model parameters, instead of a random initialization, is likely to decrease the chances of getting caught in local minima.

For the data in Table 5, the modeling approach was similar to that for the previous example (Fig. 4). The best result was obtained for $\gamma = 1.0$, with ten hidden nodes determined by the supersmoother.

Training MSE is 215.832 based on scaled data. The desired feature value is 100.69 and the predicted value is 97.3181.

For the data in Table 6, the best result was again obtained for $\gamma = 1.0$, with loading directions being initialized by linear PCR (Fig. 5). The training MSE based on the scaled input data is 0.0168. The desired feature value is 199.9 and the predicted value is 217.4902. The results of NLCR modeling are compared with those obtained by OFBNN in Table 7. This indicates that the results of NLCR modeling are comparable to those of OFBNN.

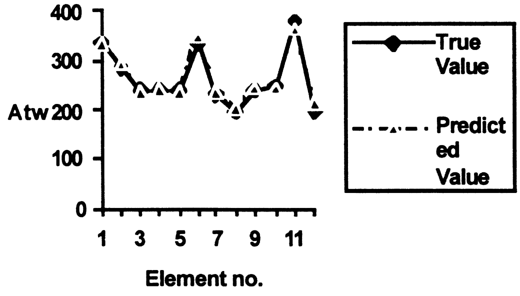

Fig. 4. Result of NLCR model for data in Table 5.

5. Conclusions and discussion

A common framework is presented for comparing neural and statistical empirical modeling methods. This framework is based on the representation of empirical models as expansion on a set of basis functions, and the realization that various methods may be obtained depending on decisions about only three criteria: the nature of the input transformation, type of basis functions, and optimization criteria for estimating the model parameters.

The insight provided by the common framework is used to develop 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 uses adaptive basis functions, a general

Table 7
Comparison of % prediction error of NLCR and OFBNN

Data set	NLCR % prediction error	OFBNN % prediction error
Table 4	0	0
Table 5	3.35	13.26
Table 6	8.80	4.17

objective function, and an efficient hierarchical training methodology. The common objective function introduces a new adjustable parameter, γ, to span the continuum of methods from NLPCR to PPR/BPN. This parameter complements the effect of the number of basis functions on the bias of the empirical model, resulting 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.

Modeling by NLCR is likely to result in maximum improvement for problems where the input variables are related to each other, and limited quantities of training data are available such as, fault diagnosis. 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.

The second example shows that the performance of NLCR is similar to that of the OFBNN method. The primary difference between the two approaches is that NLCR is a method based on linear projection, whereas OFBNN is a method based on nonlinear local projection. This implies that in NLCR, the inputs are projected as shown in Fig. 1(a), whereas OFBNN projects the inputs as shown in Fig. 1(c). The training methodology for NLCR and OFBNN is also different. Both modeling approaches are hierarchical, and train one node at a time to minimize the residual error of approximation. An important difference between the two methods is in the approach for determining the nature of the basis functions. The basis functions in NLCR adapt to the nature of the available data by smoothing the data in the space of projected inputs and output. In contrast, the basis functions in OFBNN are selected from a library of shapes. The adaptive basis functions of NLCR usually result in less number of hidden nodes in the model, and may provide greater insight into the nature of the available data than OFBNN. The overall computational complexity of the NLCR model may be less than that of the OFBNN model for problems such as molecular beam epitaxy [3]. A thorough comparison between the two methods is in progress.

Acknowledgements

Partial financial support from an Ohio State University Seed Grant, and the Air Force Summer Faculty Research Program are gratefully acknowledged.

References

[1] B.R. Bakshi, U. Utojo, Neural and Statistical Methods for Empirical Modeling: A Common Framework and Overview, Anal. Chim. Acta, submitted, 1997.

[2] B.R. Bakshi, U. Utojo, Unification of Neural and Statistical Modeling Methods that Combine Inputs by Linear Projection, Comput. Chem. Eng., (1998) in press.

[3] B.R. Bakshi, Modeling Of Materials Manufacturing Processes By Nonlinear Continuum Regression, Final Report, Summer Faculty Research Program, AFOSR, 1997.

[4] S.E Falilman, C. Lebiere, The Cascaded-Correlation Learning Architecture, Advances in Neural Information Processing Systems, vol. 2, Morgan Kaufmann, 1990, pp. 524–532.

[5] J.H. Friedman, W. Stuetzle, Projection pursuit regression. J. Am. Stat. Assoc. 76 (1981) 817–823.

[6] J.H. Friedman, Classification and Multiple Regression Through Projection Pursuit, Technical Report No. 12, Department of Statistics, Stanford University, Stanford, CA, 1985.

[7] T.R. Holcomb, M. Moran, PLS/Neural Networks, Comput. Chem. Eng. 16(4) (1992) 393–411.

[8] J.N. Hwang, M. Lay, R.D. Martin, J. Schimert, Regression modeling in back-propagation and projection pursuit learning, IEEE Trans. Neural Networks 5 (1994) 342–353.

[9] H. Martens, T. Naes, Multivariate Calibration, Wiley, New York, 1989.

[10] C.B. Roosen, T.J. Hastie, Automatic Smoothing Spline Projection Pursuit, J. Comput. Graph. Stat. 3(3) (1994) 235–248.

[11] D.E. Rumelhart, J.L. McClelland et al., Parallel Distributed Processing, vol. 1, The MIT Press, Cambridge, MA, 1986.

[12] M. Stone, R.J. Brooks, Continuum regression cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression, J. R. Stat. Soc., Ser. B. 52 (1990) 237–269.

[13] S. Wold, N. Kettaneh-Wold, B. Skagerberg, Nonlinear PLS modeling, Chemometr. Intell. Lab. Sys. 7 (1989) 53–65.

[14] S. Wold, Nonlinear Partial Least Squares Modeling II. Spline Inner Relation, Chemometr. Intell. Lab. Sys. 14(7) (1992) 1–84.

[15] C.L. Chen, Y. Cao, S.R. LeClair, Materials structure-property prediction using a self-artitecting neural network, J. Alloys Comp. 279(1) (1998) 30–38.

[16] A.G. Jackson, Z. Pawlak, S.R. LeClair, Rough sets applied to the discovery of materials knowledge, J. Alloys Comp. 279(1) (1998) 14–21.

[17] Y.H. Pao, Z. Meng, S.R. LeClair, B. Igelnik, Materials discovery via topologically-correct display of reduced-dimension data, J. Alloys Comp. 279(1) (1998) 22–29.