Pergamon

PII: S0098-1354(96)00311-0

Computers chem. Engng Vol. 21, No. 8, pp. 875–890, 1997

0098-1354/97 $17.00+0.00

Nonlinear partial least squares

E. C. Malthouse,* A. C. Tamhane and R. S. H. Mah

Northwestern University, Evanston, IL 60208, U.S.A.

(Received 8 November 1995; revised 17 June 1996)

Abstract

We propose a new nonparametric regression method for high-dimensional data, nonlinear partial least squares (NLPLS), which is motivated by projection-based regression methods, e.g. PLS, projection pursuit regression and feedforward neural networks. The model takes the form of a composition of two functions. The first function in the composition projects the predictor variables onto a lower-dimensional curve or surface yielding scores, and the second predicts the response variable from the scores. We implement NLPLS with feedforward neural networks. NLPLS often will produce a more parsimonious model (fewer score vectors) than projection-based methods. We extend the model to multiple response variables and discuss situations when multiple response variables should be modeled simultaneously and when they should be modeled with separate regressions. We provide empirical results that evaluate the performances of NLPLS, projection pursuit, and neural networks on response variable predictions and robustness to starting values. © 1997 Elsevier Science Ltd

1. Introduction

The regression problem involves modeling a function between one or more predictor variables and one or more response variables from a dataset of observations. Let $\mathbf{X}(n\times p)$ and $\mathbf{Y}(n\times q)$ be mean-centered matrices of p predictor and q response variables over n data vectors (subjects, cases, items, etc.). Denote the ith row vectors of $\mathbf{X}$ and $\mathbf{Y}$ by $\mathbf{x}_i'(1\times p)$ and $\mathbf{y}_i'(1\times q)$, respectively. The response vector $\mathbf{y}_i$ is hypothesized to be some continuous function φ of predictor vector $\mathbf{x}_i$ with additive error $\mathbf{e}_i$:

$$\mathbf{y}_i=\varphi(\mathbf{x}_i)+\mathbf{e}_i. \tag{1}$$

The elements in $\mathbf{Y}$ are assumed to take values in a continuous set. The regression problem is to construct an estimate $\hat{\varphi}$ for φ.

Many nonparametric models have been proposed for φ. Several methods that have been highly successful in practice take the following form:

$$\hat{\varphi}(\mathbf{x}_i)=\sum_{k=1}^{r} h_k(\mathbf{x}_i'\mathbf{u}_k), \tag{2}$$

where $\mathbf{u}_k$ $(p\times 1)$ is called a *loading vector* and h_k is a univariate smooth *transfer function*. We call methods that fit models of this form *projection-based regression methods*. The product $s_{ik}=\mathbf{x}_i'\mathbf{u}_k$ is called a *score* and it gives the length of the projection[1] of $\mathbf{x}_i$ onto $\mathbf{u}_k$. Each term in the sum, $h_k(\mathbf{x}_i'\mathbf{u}_k)$, is called a *ridge function*[2], because it is constant on hyperplanes orthogonal to $\mathbf{u}_k$, e.g. $\{\mathbf{x}: \mathbf{x}'\mathbf{u}_k=c\}$ for some c. Each ridge function projects the predictor variables onto a vector and relates the lengths of the projections to the response variable with the *transfer function*, h_k. Geometrically, the projection-based regression methods stack the surfaces defined by the ridge functions on top of each other to approximate φ.This paper proposes two nonlinear extensions of the projection-based regression methods, called *nonlinear partial least squares* (NLPLS). The two extensions are *sequential NLPLS* and *simultaneous NLPLS*. Sequential NLPLS generalizes the ridge functions used by the projection-based methods by projecting the predictor variables onto curve[3] $\mathbf{f}_k:\Re\rightarrow\Re^p$ instead of vector $\mathbf{u}_k$. The sequential NLPLS model has the form:

$$\hat{\varphi}(\mathbf{x}_i)=\sum_{k=1}^{r} h_k(s_{f_k}(\mathbf{x}_i)), \tag{3}$$

* To whom correspondence should be addressed.

[1] The projection of $\mathbf{x}$ onto vector $\mathbf{u}$ is given by $\text{proj}_{\mathbf{u}}\mathbf{x}=(\mathbf{x}'\mathbf{u}/\mathbf{u}'\mathbf{u})\mathbf{u}$. The expression $(\mathbf{x}'\mathbf{u}/\mathbf{u}'\mathbf{u})$ gives the length of the projection and $\mathbf{u}$ gives the direction. When $\mathbf{u}$ has unit length, $\mathbf{x}'\mathbf{u}$ gives the length of the projection; otherwise $\mathbf{x}'\mathbf{u}$ is proportional to the length of the projection.

[2] A ridge function is a function that maps $\Re^p\rightarrow\Re$ of the form $h(\mathbf{x}'\mathbf{u})$, where $\mathbf{u}$ is a $p\times 1$ vector and h is a univariate smooth transfer function.

[3] A curve in $\Re^p$ is a vector of smooth functions $\mathbf{f}(s)=(f_1(s),\ldots,f_p(s))'$ that maps $\Re$ into $\Re^p$. For example, a circle is an example of a curve in $\Re^2$, $\mathbf{f}(s)=(\cos s, \sin s)'$.

where $s_{f_k}: \Re^p \rightarrow \Re$ is called a *projection index* and is analogous to $\mathbf{x}'\mathbf{u}$ giving the location of the projection of $\mathbf{x}_i$ onto curve $\mathbf{f}_k$ in terms of arc length; h_k is a univariate smooth transfer function relating the arc lengths to the response variable.

Simultaneous NLPLS is designed for a slightly different problem than is sequential NLPLS. In problems with high-dimensional predictor variables, one is often not interested in estimating φ over its entire domain. Instead, one is only interested in estimating φ over a small subspace of $\Re^p$. A common example of this situation is when there is multicollinearity among the predictor variables, i.e. when there are dependencies among the predictor variables and the rank of matrix $\mathbf{X}$ is less than p. When this is the case, the observed predictor variables will lie approximately in a lower-dimensional *linear* subspace of $\Re^p$. The projection-based methods [equation (2)] and sequential NLPLS make no attempt to model explicitly the lower-dimensional subspace in which the predictor variables lie. In each step of the algorithm, they choose a vector $\mathbf{u}_k$ so that the lengths of the projections onto this vector will be good predictors of the response variable. Simultaneous NLPLS generalizes the philosophy behind the projection-based methods by selecting an r-dimensional surface[4] $\mathbf{f}$ in $\Re^p$ in which the predictor variables lie *and* that facilitates predicting of the response variables. The simultaneous NLPLS model is:

$$\hat{\varphi}(\mathbf{x}_i) = \mathbf{h}(\mathbf{s}_f(\mathbf{x}_i)), \tag{4}$$

where $\mathbf{s}_f$: $\Re^p \rightarrow \Re^r$ is a projection index, giving the r-dimensional coordinates of the projection of $\mathbf{x}_k$ onto $\mathbf{f}$, and $\mathbf{h}$: $\Re^r \rightarrow \Re$ is a transfer function.We also propose a multivariate version of simultaneous NLPLS that models multiple response variables. Instead of observing a single $n \times 1$ response variable $\mathbf{y}$, suppose we have n observations on q response variables $\mathbf{Y}(n \times q)$. Like the univariate version, multivariate NLPLS finds the r-dimensional surface $\mathbf{f}$ in which the predictor variables lie; it extends NLPLS by also finding the s-dimensional surface g in $\Re^q$ in which the response variables lie. Multivariate NLPLS relates the predictor variable scores to the response variable scores with a transfer function $\mathbf{h}$. The model has the form:

$$\varphi(\mathbf{x}_i) = \mathbf{g}(\mathbf{h}(\mathbf{s}_f(\mathbf{x}_i))), \tag{5}$$

where projection index $\mathbf{s}_f$ maps $\Re^p \rightarrow \Re^r$, transfer function $\mathbf{h}$ maps $\Re^r \rightarrow \Re^s$, and response variable surface $\mathbf{g}$ maps $\Re^s \rightarrow \Re^q$. Section 2 gives an overview of some commonly used projection-based regression methods. Section 3 describes the nonlinear principal components analysis (NLPCA) method, which we use to model the curves and surfaces in the NLPLS method. Section 4 describes the NLPLS models and how we estimate them. Section 5 presents a diagnostic for detecting nonlinear relationships among predictor and/or response variables. Section 6 presents the results of some empirical tests that compare NLPLS to projection-based regression methods. Section 7 summarizes our conclusions and gives directions for future research.

[4] An r-dimensional surface in $\Re^p$ is a vector of multivariate functions $\mathbf{f}(s) = (f_1(\mathbf{s}), \ldots, f_p(\mathbf{s}))'$ that maps $\Re^r$ into $\Re^p$.

2. Projection-based regression methods

The general form of a projection based regression method was given in equation (2). The purpose of this section is to give some details of the models, estimation, and approximation properties of some important projection-based regression methods to motivate our extensions.

2.1. Partial least squares

This section provides a brief overview of the Partial Least Squares (PLS) regression method. There are two PLS algorithms, PLS1 for problems with a univariate response variable ($q=1$), and PLS2 for problems with multivariate response variables ($q>1$). Sequential NLPLS is a direct generalization of PLS1 and we summarize only PLS1; see Breiman and Friedman (1994) for discussion of the PLS2 algorithm.

The PLS1 algorithm is a sequential algorithm. In the kth step of the algorithm, PLS1 extracts a single loading vector ($\mathbf{u}_k$) from the predictor variables so that the resulting scores ($\mathbf{s}_k$) are "good predictors" of the response variable residuals from the previous step. It next regresses the predictor and response variable residuals on the corresponding scores yielding regression coefficients $\mathbf{w}_k$ and v_k, respectively. Then it computes new residuals for the predictor and response variables. This process is repeated on the residual matrices until some stopping criterion is met. The steps in the PLS1 algorithm are as follows:

0. *Initialize.* Let dimension index $k=1$. Let $\mathbf{D}_0$ and $\mathbf{e}_0$ be copies of (mean-centered) $\mathbf{X}$ and $\mathbf{y}$, respectively. In subsequent iterations $\mathbf{D}_k$ and $\mathbf{e}_k$ will contain predictor and response variable residuals.

1. *Select loading vectors and scores.* Define loading vector:

$$\mathbf{u}_k = \mathbf{D}'_{k-1}\mathbf{e}_{k-1} \propto \mathrm{Cov}(\mathbf{D}_{k-1}, \mathbf{e}_{k-1}) \tag{6}$$

and score vector:

$$\mathbf{s}_k = \mathbf{D}_{k-1}\mathbf{u}_k. \tag{7}$$

The symbol $\propto$ means "proportional to." Note that the columns in $\mathbf{D}_{k-1}$ that are highly associated with $\mathbf{e}_{k-1}$ will receive large loadings relative to the columns with smaller associations.

2. *Regress predictor and response variables on scores.* The regression coefficients $\mathbf{w}_k$ and v_k are estimated with OLS:

$$\mathbf{w}_k = [(\mathbf{s}_k'\mathbf{s}_k)^{-1}\mathbf{s}_k'\mathbf{D}_{k-1}]' = \frac{\mathbf{D}'_{k-1}\mathbf{s}_k}{\mathbf{s}'_k\mathbf{s}_k} \tag{8}$$

and

$$v_k=(\mathbf{s}_k'\mathbf{s}_k)^{-1}\mathbf{s}_k'\mathbf{e}_{k-1}=\frac{\mathbf{e}_{k-1}'\mathbf{s}_k}{\mathbf{s}_k'\mathbf{s}_k}. \quad (9)$$

3. *Compute residual matrices.* The residuals are computed with:

$$\mathbf{D}_k=\mathbf{D}_{k-1}-\mathbf{s}_k\mathbf{w}_k' \text{ and } \mathbf{e}_k\mathbf{e}_{k-1}-\mathbf{s}_k v_k.$$

4. *Loop.* Increment k and return to (1) until the residual matrices $\mathbf{D}_k$ and $\mathbf{e}_k$ are sufficiently small.

The PLS predictions are given by:

$$\hat{\mathbf{y}}=\sum_{k=1}^{r}\mathbf{s}_k v_k=\sum_{k=1}^{r}\mathbf{D}_{k-1}\mathbf{u}_k v_k, \quad (10)$$

where r is the number of factors extracted. Because the process is repeated on the residuals of the predictor variables themselves instead of on the predictor variables, PLS might not appear to fit exactly the form in (2). Frank and Friedman (1993) (Section 3) discuss how the PLS algorithm can be placed into the projection-method form.

2.2. *Projection pursuit regression*

Friedman and Stuetzle (1981) first proposed the projection pursuit regression method for univariate response variables[5]. Project pursuit estimates the parameters in equation (2) with a sequential algorithm. In the kth step of the algorithm ($k=1,\ldots, r$), it selects loading vector $\mathbf{u}_k$ and transfer function h_k so that the ridge function $h_k(\mathbf{x}'\mathbf{u}_k)$ gives good predictions of the response variable residuals from the previous $k-1$ steps. The algorithm is as follows:

0. *Initialize.* Let dimension index $k=1$. Let $\mathbf{e}_0=\mathbf{y}$. In subsequent iterations, $\mathbf{e}_k$ will contain response variable residuals.
1. *Choose initial loading vector and transfer function.* Let $\mathbf{u}\in\Re^p$ be any unit-length p vector and fit a smooth curve, e.g. a cubic spline, h_k to estimate $\mathbf{e}_{k-1}$ from $\mathbf{Xu}$.
2. *Optimize.* With h_k fixed, estimate $\mathbf{u}_k$ using:

$$\mathbf{u}_k=\underset{\mathbf{u}}{\operatorname{argmin}}\sum_{i=1}^{n}(e_{k-1,i}-h_k(\mathbf{u}'\mathbf{x}_i))^2,$$

where $e_{k-1,i}$ is the ith element of the residual vector $\mathbf{e}_{k-1}$. Fit a smooth curve h_k to estimate $\mathbf{e}_{k-1}$ from $\mathbf{Xu}_k$.

3. *Compute residuals.* Compute residuals using:

$$e_{k,i}=e_{k-1,i}-h_k(\mathbf{x}_i'\mathbf{u}_k).$$

4. *Loop.* Increment k and return to (1) until the residual vector $\mathbf{e}_k$ is small.

Diaconis and Shahshahani (1984) develop some approximation theory results for models of the form in equation (2). In particular, the authors show that project pursuit can uniformly approximate arbitrary continuous functions on $[0, 1]^p$ and establish necessary and sufficient conditions for exact representation of φ with a finite sum of ridge functions (i.e. finite r). They give the following examples of functions that require infinite r for an exact representation:

$$\varphi(x_1,x_2)=e^{x_1x_2} \text{ and } \varphi(x_1,x_2)=\sin(x_1x_2). \quad (11)$$

2.3. *Feedforward neural networks*

Feedforward neural networks with one hidden layer can also be written in the form of equation (2). A three-layer neural network with p input nodes, r hidden nodes, and one output node fits a model of the form:

$$\hat{\varphi}(\mathbf{x})=\sum_{k=1}^{r}\sigma(\mathbf{x}'\mathbf{u}_k)v_k, \quad (12)$$

where $\sigma(\cdot)$ is a fixed transfer function with a sigmoidal shape, e.g. the scaled logistic function $\sigma(s)=(1-e^{-s})/(1+e^{-s})$. Weight values $\mathbf{u}_k(p\times 1)$ and v_k ($k=1,\ldots, r$) are selected so that the following least squares objective function is minimized:

$$\min_{\mathbf{u}_1,\ldots,\mathbf{u}_r,v_1,\ldots,v_r}\sum_{i=1}^{n}\|\mathbf{y}_i-\sum_{k=1}^{r}\sigma(\mathbf{x}_i'\mathbf{u}_k)v_k\|^2.$$

By defining $h_k(s)=v_k\sigma(s)$, a feedforward neural network is recognized as a projection-based regression method [equation (2)]. Neural networks are different from all other projection-based regression methods discussed here in two main ways: (1) parameters (weights) $\mathbf{u}_k$ and v_k ($k=1,\ldots, r$) are estimated simultaneously vis-à-vis with a sequential algorithm that estimates each ridge function separately; and (2) vector $\mathbf{u}_k$ is not constrained to have a unit length. By changing the length of $\mathbf{u}_k$, one can stretch or compress the sigmoidal function $\sigma(\cdot)$; v_k is a scaling factor. Cybenko (1989) has shown that these networks can represent arbitrary continuous functions under weak conditions. The approximation improves with more hidden nodes.

2.4. *Extensions of PLS*

Several authors have extended PLS to model the general regression problem defined in equation (1). These methods are designed for problems with a low observation-to-variable ratio, multicolinearity among the predictors, and a nonlinear relationship between predictor and response variables. Partial least squares is not well suited for these problems because its model is nearly linear[6]. Projection pursuit and neural networks are not well suited either because they must estimate too many parameters with too few observations (Frank, 1994). The PLS extensions overcome the limitations of PLS by modeling the transfer functions nonparametrically. They estimate fewer parameters than projection pursuit and neural networks by first selecting the loading vectors with PLS or principal components analysis (PCA) (Mardia et al., 1979) and, with the loading

[5] See Hwang et al. (1995) for discussions of how projection pursuit can model multivariate response variables.

[6] See Frank and Friedman (1993) for discussion of why the PLS model is "nearly" linear.

vectors fixed, then optimizing the choice of transfer function. Frank (1994) proposes "Neural Networks based on PCR and PLS components nonlinearized by Smoothers and Splines" for univariate response variables. Several similar extensions have been proposed for PLS2 including quadratic PLS with two blocks (Wold et al., 1989), neural network PLS (Qin and McAvoy, 1992) and spline PLS (Wold, 1992).

3. Nonlinear principal components analysis

The NLPLS generalizes the projection methods by projecting the predictor variables onto curves or surfaces instead of one-dimensional vectors. Several methods have been proposed to extract a curve or surface from a set of p-dimensional observations including Nonlinear Principal Components Analysis (NLPCA) (Kramer, 1991), which estimates a curve or a surface passing "through the middle" of the observations using least squares:

$$\min_{\mathbf{f},\mathbf{s}_\mathbf{f}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{f}(\mathbf{s}_\mathbf{f}(\mathbf{x}_i))\|^2, \tag{13}$$

where $\mathbf{s}_\mathbf{f}$ maps from $\Re^p$ to $\Re^r$ and is called a *projection index*. Function $\mathbf{f}$ is a vector of smooth functions mapping from $\Re^r$ to $\Re^p$ ($r<p$) called an r-dimensional surface; when $r=1$, $\mathbf{f}$ is usually called a curve. The projection index, $\mathbf{s}_\mathbf{f}$ reduces the dimension of $\mathbf{x}$. The composition of functions $\mathbf{f}(\mathbf{s}_\mathbf{f}(\mathbf{x}_i))$ gives the p-dimensional coordinates of the projection[7] of $\mathbf{x}_i$ onto a curve or surface $\mathbf{f}$. Functions $\mathbf{s}_\mathbf{f}$ and $\mathbf{f}$ are modeled nonparametrically. The NLPCA models each with three-layer neural networks. A five-layer neural network is used to model the composition of functions $\mathbf{f}$ and $s_\mathbf{f}$. Layers 1–3 model function $\mathbf{s}_\mathbf{f}$ and layers 3–5 model function $\mathbf{f}$. The five-layer NLPCA network has p nodes in the input layer, r nodes in the third (bottleneck) layer, and p nodes in the output layer. The top part of Fig. 1 shows an example of an NLPCA network.

4. Nonlinear PLS

4.1. Simultaneous univariate NLPLS

The general form of the simultaneous univariate NLPLS model is:

$$\hat{\varphi}(\mathbf{x}_i) = \mathbf{h}(\mathbf{s}_\mathbf{f}(\mathbf{x}_i)), \tag{14}$$

where $\mathbf{s}_\mathbf{f}$: $\Re^p \rightarrow \Re^r$ is a projection index, giving the r-dimensional coordinates of the projection of $\mathbf{x}_k$ onto curve $\mathbf{f}$, and $\mathbf{h}$: $\Re^r$ to $\Re$ is a transfer function. The curve, projection index, and transfer function in this model are selected to minimize the following objective function:

$$\min_{\mathbf{f},\mathbf{s}_\mathbf{f},h} \sum_{i=1}^{n} [\|\mathbf{x}_i - \mathbf{f}(\mathbf{s}_\mathbf{f}(\mathbf{x}_i))\|^2 + \|y_i - h(\mathbf{s}_\mathbf{f}(\mathbf{x}_i))\|^2]. \tag{15}$$

[7] Malthouse (1995) shows that since NLPCA models $\mathbf{s}_\mathbf{f}$ with a continuous function, the point on curve $\mathbf{f}$ indexed by $s_\mathbf{f}(\mathbf{x})$ is not always the point on $\mathbf{f}$ that is closest to $\mathbf{x}$, and is hence not a "projection" in a strict sense.

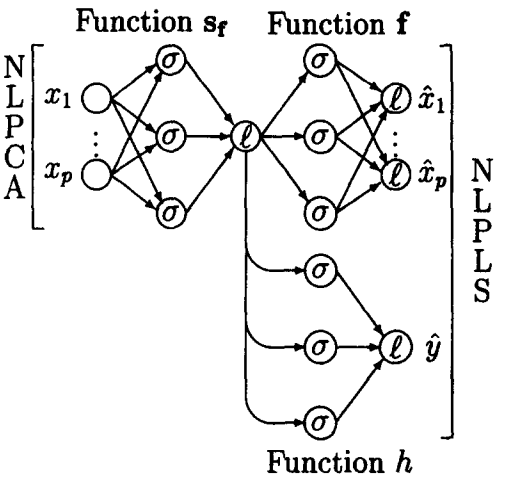

Fig. 1. Neural network architecture of NLPLS. A linear function is denoted by ℓ, and a sigmoidal function is denoted by σ.

Just as PLS1 regresses $\mathbf{X}$ and $\mathbf{y}$ on its scores [equations (8) and (9)], the two terms in the NLPLS objective function regress $\mathbf{X}$ and $\mathbf{y}$ on the NLPLS scores. The first term is the NLPCA objective function (13), which specifies that the dimension of the predictor variables be reduced. The second term regresses $\mathbf{y}$ on the scores. Without the first term, the objective function (15) could be minimized by fixing h to be the identity function ($h(s)=s$), setting $r=1$, and training $\mathbf{s}_\mathbf{f}$ to estimate φ, which would essentially be a three-layer neural network.

The NLPLS selection of curve $\mathbf{f}$ attempts to generalize the PLS selection of loading vectors in equation (6). NLPLS simultaneously estimates functions $\mathbf{f}$, $\mathbf{s}_\mathbf{f}$, and h to minimize objective function (15), and the score values therefore must be good predictors of *both* $\mathbf{X}$ and $\mathbf{y}$. Consequently, the curve extracted from the predictor variables and its parameterization need not to be the same as the curve extracted by NLPCA.

We implement NLPLS by modeling the terms in equation (15) with a five-layer neural network. Fig. 1 shows an example of an NLPLS network. The top part of the network (labeled "NLPCA") is an NLPCA network (simultaneous) and was discussed in Section 3. The bottom part of the network (labeled "Function h") models the transfer function h, which predicts y. The inputs to the nodes in the bottom part of the network are from the bottleneck node(s) in layer 3 of the NLPCA network. Layers 1, 2, and 3 (labeled "Function $\mathbf{s}_\mathbf{f}$") model function $\mathbf{s}_\mathbf{f}$. Thus the activation of the node in the bottleneck layer (layer three) gives the coordinates of the "projection" of $\mathbf{x}_i$ onto $\mathbf{f}$. Layers 3, 4, and 5 in the top half of the network (labeled "Function $\mathbf{f}$") model curve/surface $\mathbf{f}$. Layers 3, 4, and 5 in the bottom half of the network (labeled "Function h") model the NLPLS transfer function h. Direct ("linear bypass") connections are allowed between layers 1 and 3 and layers 3 and 5, but are not allowed to cross layer 3. See Section 5 for a discussion of how to pick r.

The NLPLS has the same approximation properties as

three-layer neural networks since three-layer neural networks are a special case of equation (14). When $r=p$ and $\mathbf{s_f(x)=f(x)=x}$ (identity mapping), function $\mathbf{h}$ is a three-layer neural network that maps $\mathbf{X}$ to $\mathbf{y}$.

4.2. *Sequential univariate NLPLS*

One reason for the success of the projection-based regression methods[8] is that they estimate their models in a sequential manner (Friedman and Stuetzle, 1981). The ridge functions in (2) are estimated sequentially to give the best fit to the response variable residuals from the previous step. Sequential NLPLS is a direct generalization of this approach and fits a model of the form:

$$\hat{\varphi}(\mathbf{x}_i)=\sum_{k=1}^{r} h_k(s_{\mathbf{f}_k}(\mathbf{x}_i)), \tag{16}$$

where $s_{\mathbf{f}_k}:\Re^p\rightarrow\Re$ is called a *projection index* giving the location of the projection of $\mathbf{x}_i$ onto curve $\mathbf{f}_k$ in terms of arc length and h_k is a univariate smooth transfer function relating the arc lengths to the response variable. The projection-based methods project the predictor variables onto vectors whereas sequential NLPLS projects the predictor variables onto curves. Sequential NLPLS find the projection indices, curves, and transfer functions with the following algorithm:

0. *Initialize.* Let dimension index $k=1$. Let $\mathbf{e}_0$ be a copy of (mean-centered) $\mathbf{y}$.
1. *Select curve and parameterization.* Define curve $\mathbf{f}_k$, projection index $s_{\mathbf{f}_k}$, and transfer function k_k to minimize the simultaneous univariate NLPLS objective function (15). This is equivalent to fitting a simultaneous univariate NLPLS model with $r=1$.
2. *Compute response-variable residuals.*

$$\mathbf{e}_{ki}=\mathbf{e}_{k-1,i}-\mathbf{f}_k(s_{\mathbf{f}_k}(\mathbf{x}_i)).$$

3. *Loop.* Increment k and return to (1) until the residual vector $\mathbf{e}_k$ is sufficiently small.

The sequential NLPLS model shares the approximation properties of projection pursuit since the projection pursuit model is a special case of (16). The projection indices $s_{\mathbf{f}}$ can model linear functions, e.g. the $\mathbf{u}_k$ in (2), and the transfer functions h_k can model continuous functions from $\Re$ to $\Re$.

4.3. *Multivariate NLPLS*

The model for simultaneous multivariate NLPLS is:

$$\hat{\varphi}(\mathbf{x}_i)=\mathbf{g}(\mathbf{h}(\mathbf{s_f}(\mathbf{x}_i))), \tag{17}$$

where projection index $\mathbf{s_f}$ maps $\Re^p\rightarrow\Re^r$, transfer function $\mathbf{h}$ maps $\Re^r\rightarrow\Re^s$, and response variable surface $\mathbf{g}$ maps $\Re^s\rightarrow\Re^q$. The dimensions of the reduced-order space for the predictor and response variables are r and s, respectively. The parameters in this model are selected to minimize the following objective function:

$$\min_{\mathbf{f},\mathbf{s_f},\mathbf{h},\mathbf{g},\mathbf{t_g}} \sum_{i=1}^{n} [\|\mathbf{x}_i-\mathbf{f}(\mathbf{s_f}(\mathbf{x}_i))\|^2+\|\mathbf{t_g}(y_i)-\mathbf{h}(\mathbf{s_f}(\mathbf{x}_i))\|^2+\|\mathbf{y}_i-\mathbf{g}(\mathbf{t_g}(\mathbf{y}_i))\|^2]. \tag{18}$$

The first and last terms in equation (18) are NLPCA objective functions [equation (13)], which specify that the dimensions of the predictor and response variables be reduced. The second term specifies that function $\mathbf{h}$ relate the predictor variable scores to the response variable scores as closely as possible.

Malthouse (1995) shows that the intrinsic dimension of the predictor variables should be greater than or equal to the intrinsic dimension of the response variables, i.e. $r \geq s$. The intuition behind this result is that if the response variables are assumed to be determined by some function φ of the predictor variables, the dimension of the intrinsic domain of φ must be at least as great as the dimension of the range of φ.

We implement multivariate NLPLS with a five-layer neural network similar to the simultaneous univariate NLPLS network. Fig. 2 shows an example of an NLPLS network. The NLPLS models functions $\mathbf{s_f}$, $\mathbf{f}$, $\mathbf{t_g}$, $\mathbf{g}$, and $\mathbf{h}$ with three-layer, feedforward neural networks. The top part of the network (labeled Function $\mathbf{s_f}$ and Function $\mathbf{f}$) is an autoassociative NLPCA network and minimizes the first term in the objective function (18). The bottom part of the network (labeled Function $\mathbf{t_g}$ and Function $\mathbf{g}$) is also an autoassociative NLPCA network and minimizes the last term in the objective function (18). The middle network minimizes the middle term in the objective function (18).

Feedforward neural network simulators are programmed to solve the least squares problem that minimizes the squared difference between observed and predicted response variables over a training set. The middle term in the objective function ($\|\mathbf{t_g}(\mathbf{y}_i)-\mathbf{h}(\mathbf{s_f}(\mathbf{x}_i))\|^2$) does not correspond to supervisory training because it minimizes the differences between the outputs of s pairs of nodes rather than between the outputs of nodes and fixed target values. Networks with this additional term in their objective function can be trained with a standard feedforward network simulator by including s *difference* nodes (labeled on Fig. 2) (Briesch and Malthouse, 1994). The purpose of a difference node is to minimize the difference between the outputs of two nodes and thus to make them as equal as possible. Difference nodes are treated as output nodes with a target value of 0 for all training vectors. The two nodes whose output difference is to be minimized have arcs feeding into the difference node, one with constant weight +1 and the other with constant weight −1. After the objective function is minimized, the parts of the network modeling $\mathbf{t_g}$, $\mathbf{f}$, and difference nodes can be discarded and the output of $\mathbf{h}(\mathbf{s_f}(\mathbf{x}))$ is given to $\mathbf{g}$ to make response-variable predictions as given in equation (17).

[8] This excludes the feedforward neural networks introduced in Section 2.3, which are trained with global nonlinear optimization algorithms.

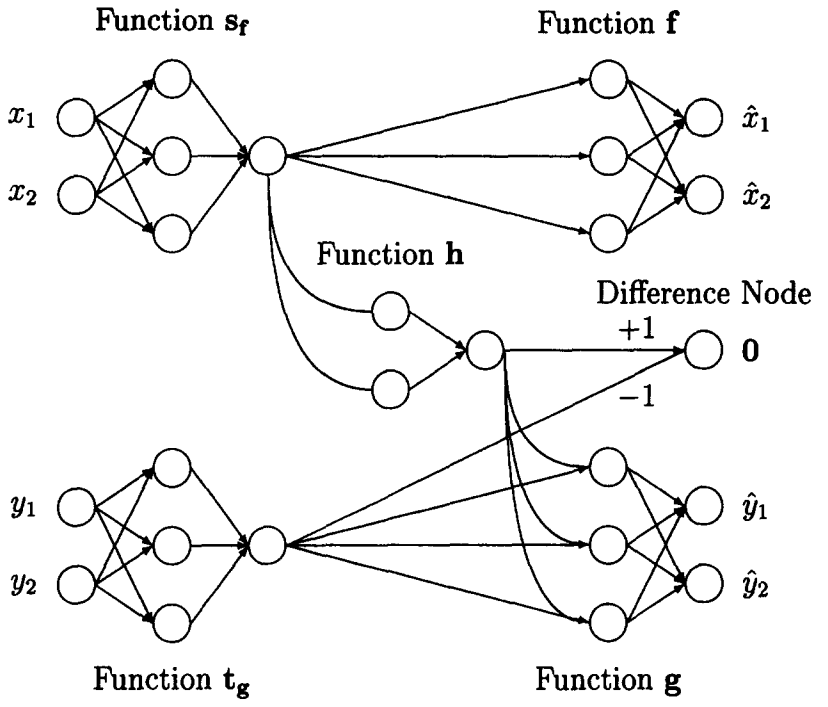

Fig. 2. Multivariate NLPLS network architecture.

5. Diagnostics

This section proposes diagnostics to answer the following questions. (1) Are simultaneous NLPLS (univariate and multivariate) appropriate for this problem? (2) How should one pick values of r and s in simultaneous NLPLS models? In this section, we show that the answers to both questions depend on knowing the intrinsic linear and nonlinear dimensions of the predictor and response variable spaces. The problem of finding the *linear* dimension of a set of multivariate observations, sometimes called the number of factors problem, has been thoroughly studied and, according to Horn and Engstrom (1979), no fewer than 50 tests have been proposed. The *scree plot* method (Cattell, 1966) has gained wide acceptance and is available in most commercial statistical software packages. It uses the eigenvalues from a principal components analysis to determine the number of factors to keep. If $\mathbf{X}$ is a mean-centered matrix of observations and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ are the eigenvalues of $\mathbf{X}'\mathbf{X}$, then λ_j gives the variance accounted for by the jth principal axis (Mardia et al., 1979). A scree plot is a plot of λ_j against j; Fig. 3 shows the scree plot of the predictor variables of an example that will be discussed later in Section 6.3. The problem is to determine which principal directions contain useful information and which contain noise. The term *scree* refers to the rubble at the base of a mountain and a scree plot will ideally look like a mountain. The real factors have eigenvalues on the steep slope of the scree plot making up the mountain and the factors containing noise have eigenvalues making up the rubble. To determine the number of factors, look for an "elbow" in the plot, which marks the beginning of the scree. In the example, the elbow for principal components analysis (PCA) is located around $r=2$ or $r=3$. The observations appear to lie on a surface of dimension $r=1$, since components 2, 3,... account for hardly any variance.

Simultaneous NLPLS models relationships among predictor and/or response variables. It is therefore well-suited for problems where the predictor and/or response variables lie in a nonlinear subspace. When this is not the case, simultaneous NLPLS offers no advantage over other methods. To determine if simultaneous NLPLS should be used on a problem, compare the number of nonlinear dimensions required to describe the locations of the observations with the number of linear dimensions. If the number of nonlinear dimensions is smaller

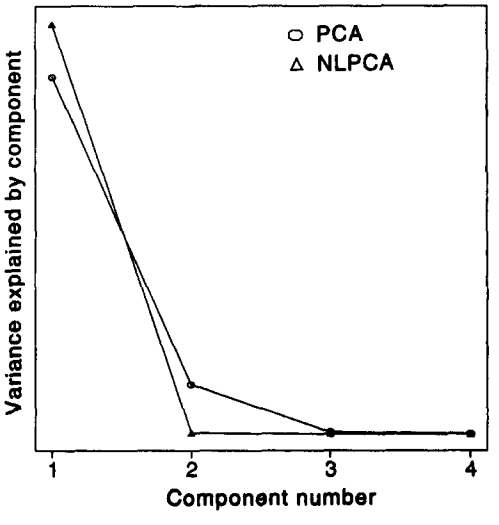

Fig. 3. Scree plot for composite materials example.

than the number of linear dimensions, the observations form a surface that is not linear; if the two are the same, the surface is linear and other regression methods should be used. Continuing the example given above, because the nonlinear dimension is smaller than the linear dimension, simultaneous NLPLS might have an advantage over projection based methods. Since the predictor variables lie on a curve, $r=1$ bottleneck nod is required for the predictor variables. The results of an analysis of these data are given in Section 6.3.

6. Examples

This section presents the results from some empirical tests that compare the performances of the NLPLS methods with three-layer neural networks and project pursuit. We chose empirical problems with different characteristics to juxtapose the performances of these methods. Section 6.1 presents the "surface problem," which compares univariate simultaneous and sequential NLPLS on their ability to model regression functions with three-dimensional predictor variables sampled from a two-dimensional surface. Section 6.2 presents the "surface problem with multiple response variables," which compares simultaneous univariate NLPLS with multivariate NLPLS on a surface problem. Section 6.3 presents the "composite materials" problem, which compares the performances of the methods on a high-dimensional problem with $p=466$ and $q=2$. We compare the methods on the following criteria:

1. *Function approximation.* We compute the coefficient of determination (R^2) of the predictions from each method to measure how well the methods approximate a given response surface.

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{q}(y_{ij}-\hat{y}_{ij})^2}{\sum_{i=1}^{n}\sum_{j=1}^{q}(y_{ij}-\bar{y}_j)^2}, \quad (19)$$

   where $\hat{y}_{ij}$ is the predicted value of y_{ij} (jth element of vector $\hat{\varphi}(\mathbf{x}_i)$), and $\bar{y}_j$ is average of the elements in the jth column of matrix $\mathbf{Y}$ ($\bar{y}_j = \sum_{i=1}^{n} y_{ij}/n$).
2. *Robustness to starting values.* We compare the methods on whether the R^2 values change with different starting values.

All simulated data sets were generated in S-Plus. The NLPLS, NLPCA, and three-layer neural networks were implemented using a neural network simulation package[9] developed by the first author in the C programming language. The package includes a network descriptor language that allows the user to define any feedforward neural network architecture, the limited-memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) (Liu and Nocedal, 1989) non-linear optimization routine for determining the weight values, and routines for computing summary statistics, function estimates, and residuals. To estimate projection pursuit models, we use the S-Plus function **ppreg**.

6.1. Surface problems

This section presents the results of nine mathematical examples that are designed to compare the performances of the two univariate NLPLS algorithms, three-layer neural networks, and projection pursuit. The examples all have $p=3$ predictor variables and $q=1$ response variables. The predictor variables were sampled from an elliptic paraboloid:

$$\mathbf{x} = \mathbf{f}(s_1, s_2) = \begin{pmatrix} s_1 \\ s_2 \\ s_1^2 + s_2^2 \end{pmatrix}. \quad (20)$$

The response variable is a function of the score values. We used three different transfer functions and added one of three different amounts of noise to the functions to give nine total examples. Without noise, the predictor (μ_{ij}) and response (υ_i) variables for the three problems were generated as follows:

- *Linear.* Let score $s_{ij} \sim U[-1,1], i=1,\ldots,200$, and $j=1,2$. Let $\mu_i = \mathbf{f}(s_{i1}, s_{i2})$ using $\mathbf{f}$ from equation (20), and $\upsilon_i = a_{i1} + s_{i2}$.
- *Quadratic.* Let score $s_{ij} \sim U[-1,1], i=1,\ldots,200$, and $j=1,2$. Let $\mu_i = \mathbf{f}(s_{i1}, s_{i2})$ using $\mathbf{f}$ from equation (20), and $\upsilon_i = a_{i1}^2 + s_{i2}^2$.
- *Exponential.* Let score $s_{ij} \sim U[-1,1], i=1,\ldots,200$, and $j=1,2$. Let $\mu_i = \mathbf{f}(s_{i1}, s_{i2})$ using $\mathbf{f}$ from equation (20) and $\upsilon_i = \exp(s_{i1}, s_{i2})$.

The first subproblem is noiseless, with $\mathbf{x}_i = \mu_i$ and $y_i = \upsilon_i$. We added Gaussian noise to μ_i and υ_i in the second and third subproblems, i.e. $\mathbf{x}_i = \mu_i + \delta_i$ and $y_i = \upsilon_i + \epsilon_i$, where δ_i (3×1) and ϵ_i (1×1) are normally distributed noise vectors. The signal-to-noise (S/N) ratios in the second and third problems are 8 and 4, respectively, e.g. when $S/N = 4, \delta_{ij} \sim N(0, \sigma_j^2/4)$, where σ_j^2 is the sample variance of the jth element of μ_i.

The transfer functions increase in difficulty. There is a linear relationship between the predictor and response variables in the linear problem; thus linear regression, all of the projection-based regression methods in Section 2, and the NLPLS methods are expected to do well. There is an additive[10] relationship between the predictor and response variables in the quadratic problem and all of the nonlinear projection-based regression methods should do well. Likewise, both the simultaneous and sequential NLPLS methods should do well. The exponential problem is the most difficult since the transfer function is a Diaconis function [equation (11)] and cannot be represented exactly by a sum of finite number of ridge functions; thus the nonlinear extensions of PLS described in Section 2.4 cannot represent it exactly because r must be at most p in these models. Sequential NLPLS should require more terms than it does for the

[9] The package is available from http://www.kellogg.nwu.edu/faculty/malthous.

[10] An *additive model* (Thisted, 1988, p. 229) has the form $\sum_{j=1}^{p} h_j(x_{ij})$, where $h_j(\cdot)$ maps $\Re \rightarrow \Re$.

quadratic problem to give a good approximation. Three-layer neural networks, projection pursuit, and simultaneous NLPLS, should all do well.

Since there are *nonlinear* relationships among the predictor variables, simultaneous NLPLS will require fewer factors than the projection-based regression methods. The predictor variables can be described by two nonlinear factors, while three linear factors are required to describe them. In this sense, the simultaneous NLPLS models will be more parsimonious than the projection-based regression models.

We fitted three-layer neural networks (labeled NN), simultaneous NLPLS (labeled Sim), and $r=1$, 2, 3 sequential NLPLS (labeled Seq1, Seq2, and Seq3) models for each of the nine examples with twenty different sets of starting values. Thus we trained 3 (functions) $\times 3$ (noise levels) $\times 5$ (Seq1, Seq2, Seq3, Sim, NN) $\times 20$ (starting values) $=900$ networks to give the results in this section. We also fitted projection pursuit (labeled PPR) models with $r=1,\ldots,5$ and selected the model with the largest training R^2 value. We show the R^2 values from these runs in Figs. 4–6. Boxplots with the word "Linear" in their title give results for problems with linear transfer functions, "Quadratic" for quadratic transfer functions, and "Exp" for exponential transfer functions. The abbreviation "Tr" indicates results from the training data and "CV" indicates cross-validation data. The first column of box plots shows the training R^2 values and the second column shows the cross-validation (CV) R^2 values. Some of the Sequential NLPLS runs produced negative R^2 values and we have omitted these runs from the plots. In cases with negative R^2 values, we indicate the number of runs in parentheses at the bottom of the plot. The R^2 values can be negative when a model overfits the data. If a model describes all the stochastic variation in the data and fails to extract the underlying surface, the predictions on a cross-validation data set can be poor. Note the following from the box plots:

- *Sequential NLPLS overfits data.* In most cases, the sequential NLPLS training R^2 values increase as r increases, while the cross-validation R^2 values decrease. In several problems, the sequential NLPLS training R^2 values are better than the R^2 values from the other methods, but the other methods have larger cross-validation R^2 values. This suggests that sequential NLPLS has a tendency to overfit the data.
- *Curves picked to predict y.* The NLPLS selection of curves attempts to generalize the PLS selection of loading vectors in (6). NLPLS simultaneously estimates functions $\mathbf{f}$, $\mathbf{s}_r$ and h to minimize the objective function (15) and the score values must be good predictors of *both* $\mathbf{X}$ and $\mathbf{y}$. Consequently, the curve extracted from the predictor variables need not be the same as the curve extracted by NLPCA. In Fig. 7 we plot the curves f_1, f_2, f_3, and f_4 from one of the sequential NLPLS models that gave a good fit to the data. The NLPLS selects each curve to estimate a different vector, since residuals are used in subsequent iterations. The first three curves ($k=1$, 2, 3) pick nearly vertical slices through the paraboloid, while the fourth curve is very nonlinear and has a horizontal "N" shape.
- *First factor in sequential NLPLS tries to do it all.* The training box plots for the second and third sequential NLPLS factors (Seq2 and Seq3) do not seem to be better than the results with only one factor (Seq1). Fig. 8 shows the curve extracted for the first factor of one of the sequential NLPLS models that has small R^2 values for the $r=2$ and three-factor solutions. The curve tries to wiggle around throughout the entire domain of the regression function. We suspect that the response variable residuals from this model are highly variable and difficult to model with subsequent factors.
- *Large variance of sequential NLPLS R^2 values.* The sequential NLPLS boxes are almost always substantially longer than the boxes for the other methods (indicating high variation), and sequential NLPLS often produces extreme R^2 values, e.g. there is a cross-validation R^2 value for Seq3 in the quadratic problem with SN=4 that is approximately -80. We conclude that sequential NLPLS is highly sensitive to starting values.
- *NN, simultaneous NLPLS, and PPR are equally good.* The three-layer neural network, simultaneous NLPLS, and projection pursuit solutions are equally good for the problems studied here. The simultaneous NLPLS and three-layer neural network models seem to be equally robust to starting values, since the boxes are roughly the same size in most cases. One of the simultaneous NLPLS models for the linear problem with SN=4 produced a poor model with $R^2 \approx 0.1$.

6.2. *Surface problems with multivariate response variables*

The difference between multivariate NLPLS and the univariate NLPLS methods is that multivariate NLPLS can model multiple response variables ($q>1$) simultaneously. This section presents some empirical results that attempt to evaluate whether multiple response variables should be modeled with separate univariate models or with multivariate NLPLS. Several recent papers, including Frank and Friedman (1993), Garthwaite (1994), Brooks and Stone (1994) and Breiman and Friedman (1994) have discussed when PLS2 and related linear methods should be used instead of separate univariate models. Breiman and Friedman (1994, p. 32) find that "the best of the multiple response procedures considered [in their study, not including two-block PLS] can provide large gains in the expected prediction accuracy (for each response), over separate single response regressions, with surprisingly little risk of making things worse."

The multivariate NLPLS approach also has several potential problems that must be balanced with the benefits mentioned above:

1. The size of parameter space grows exponentially with the number of parameters to be fit. As the number of

response variables grows, the number of parameters that must be estimated also grows. Finding a global minimum can become more difficult as the size of the search space grows. This is particularly true with the three-layer feedforward neural network and NLPLS methods that estimate all parameters simultaneously.

2. The response variable score values must be chosen to predict all the response variables. When the response

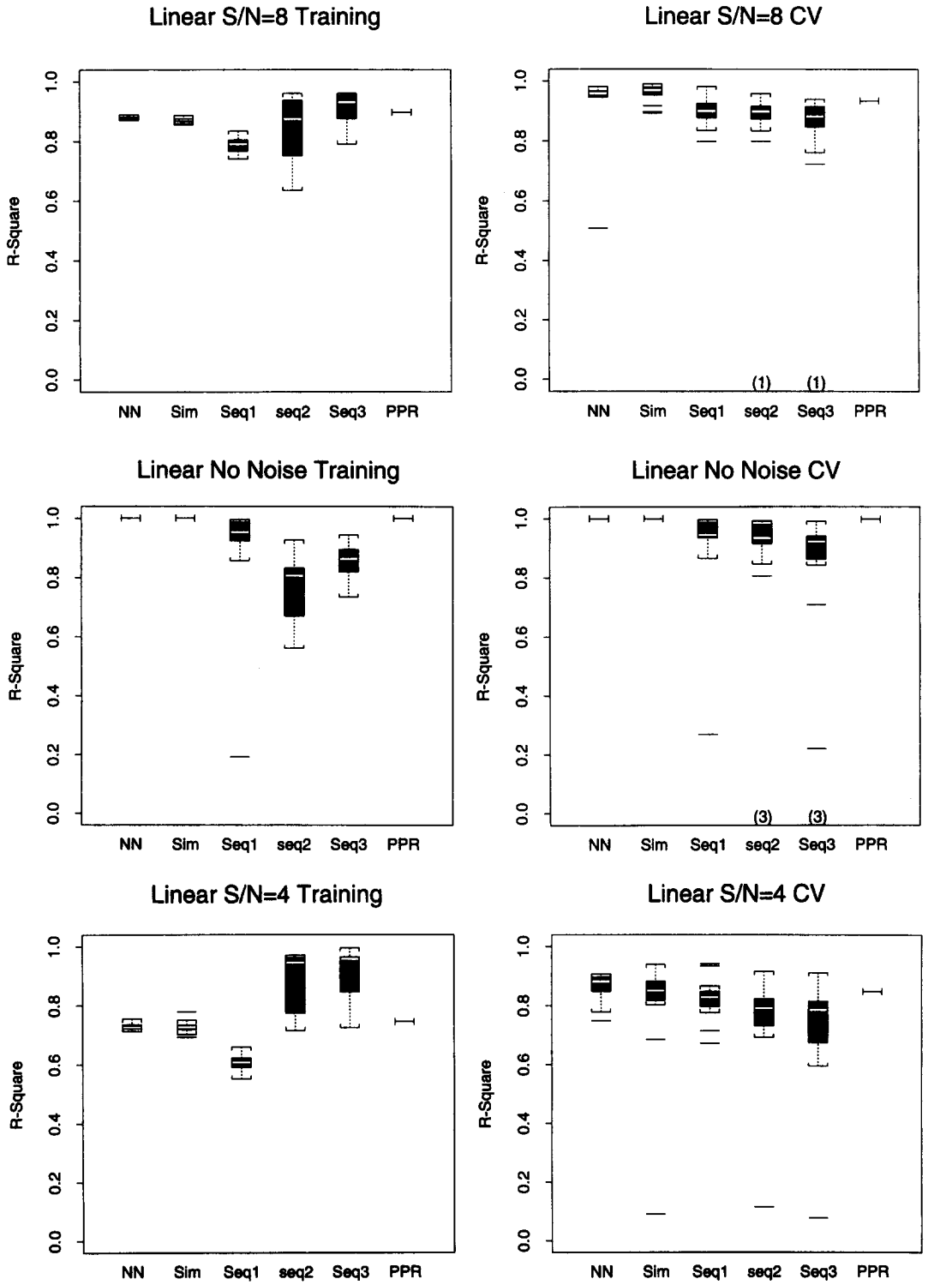

Fig. 4. Box plots of R^2 values for surface problem with univariate response variable and linear transfer function using different starting values. Abbreviations: CV, cross-validation; Sim, simultaneous NLPLS; Seq, sequential NLPLS. The numbers in parentheses give the number of runs generating negative R^2 values.

variables do not have a nonlinear relationship, the simultaneous model can require more nonlinear factors than the separate models.

This example extends the surface problem discussed in Section 6.1 so that simultaneous univariate NLPLS and multivariate NLPLS can be compared. In addition to having multiple predictor variables, the problems in this section also have multiple response variables which are nonlinearly related to each other. We sampled predictor

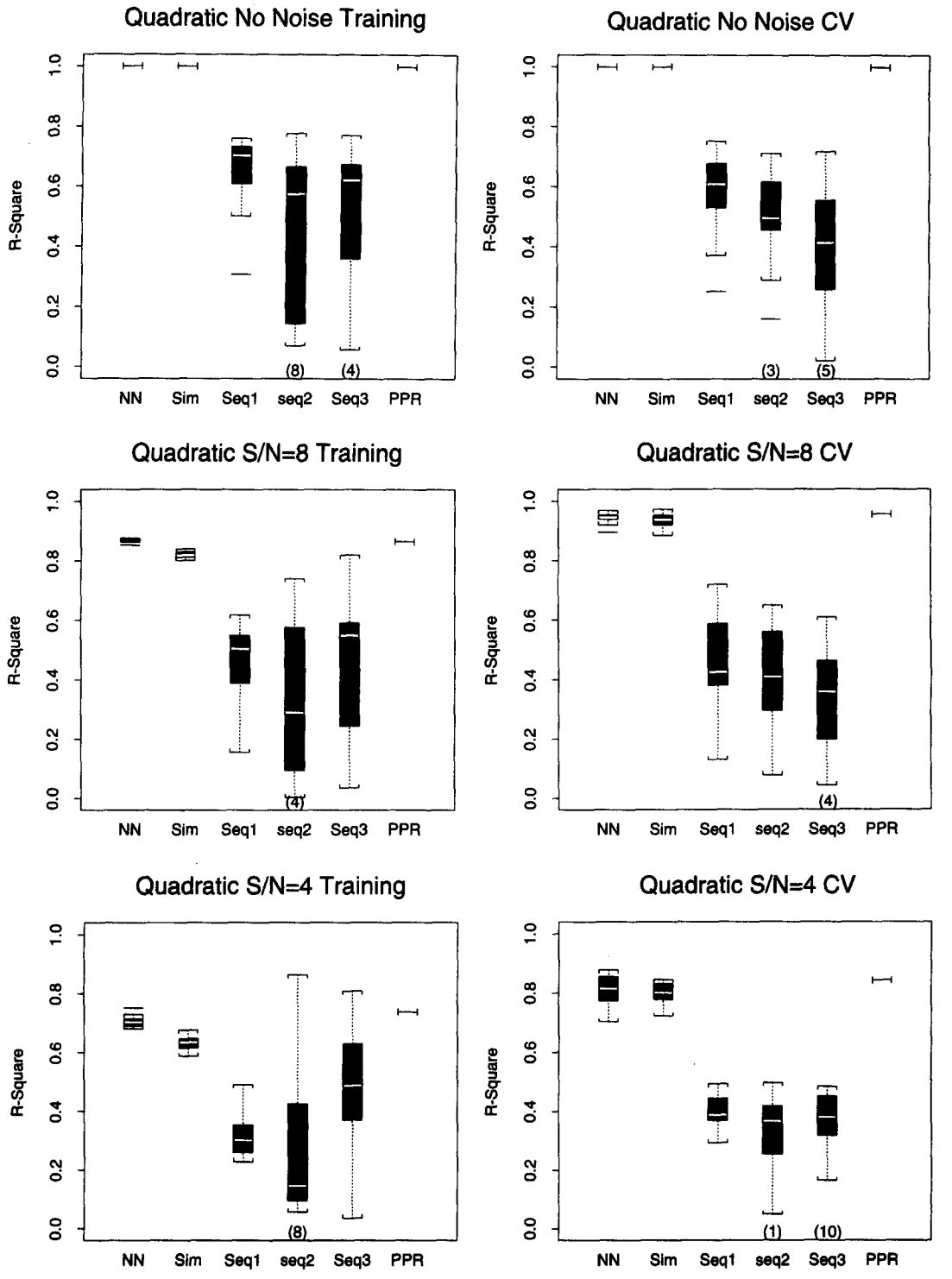

Fig. 5. Box plots of R^2 values for surface problem with univariate response variable and quadratic transfer function using different starting values. Abbreviations: CV, cross-validation; Sim, simultaneous NLPLS; Seq, sequential NLPLS. The numbers in parentheses give the number of runs generating negative R^2 values.

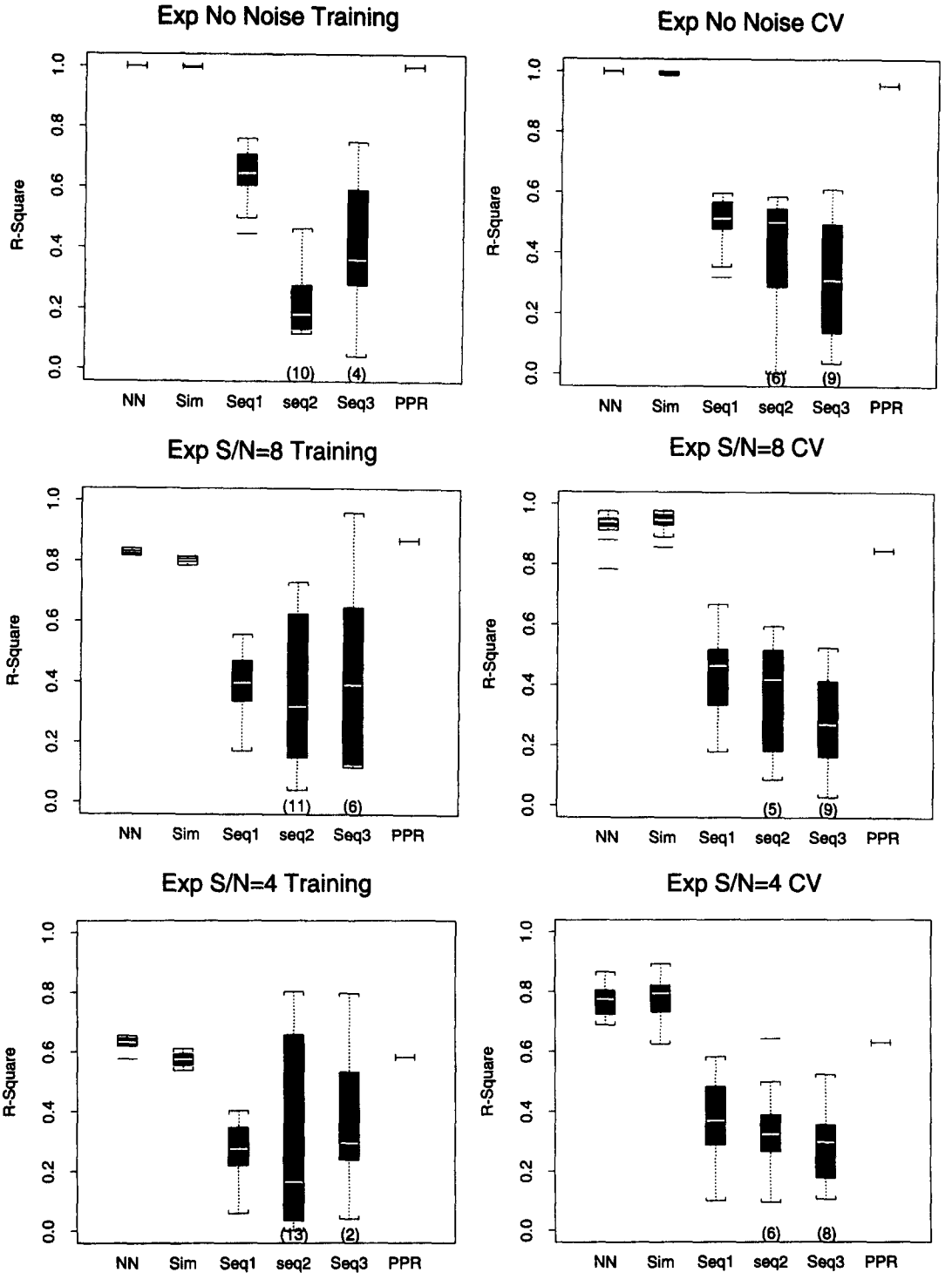

Fig. 6. Box plots of R^2 values for surface problem with univariate response variable and exponential transfer function using different starting values. Abbreviations: CV, cross-validation; Sim, simultaneous NLPLS; Seq, sequential NLPLS. The numbers in parentheses give the number of runs generating negative R^2 values.

variable scores $s_{ij} \sim U[-1,1]$, $i=1,...,200$, $j=1,2$ and defined the response variable scores t_{ij} as follows:

$$\begin{pmatrix} t_{i1} \\ t_{i2} \end{pmatrix} = \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} s_{i1} \\ s_{i2} \end{pmatrix}.$$

The relationships between observed variables and scores are elliptic paraboloids:

$$\begin{pmatrix} \mu_{i1} \\ \mu_{i2} \end{pmatrix} = \begin{pmatrix} s_{i1} \\ s_{i2} \\ s_{i1}^2 + s_{i2}^2 \end{pmatrix} \text{ and } \begin{pmatrix} u_{i1} \\ u_{i2} \end{pmatrix} = \begin{pmatrix} t_{i1} \\ t_{i2} \\ t_{i1}^2 + t_{i2}^2 \end{pmatrix}.$$

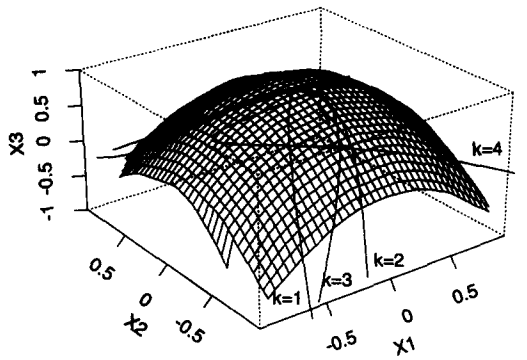

Fig. 7. Sequential NLPLS curves for the paraboloid example.

There is a nonlinear relationship among both the predictor (μ) and response (υ) variables, and NLPLS should require two NLPLS factors to model the underlying surface. The relationship between the predictor and response variable score vectors is linear.

Three subproblems were generated. The first subproblem is noiseless with $\mathbf{x}_i=\mu_i$ and $\mathbf{y}_i=\upsilon_i$. Gaussian noise was added to μ and υ in the second and third subproblems, i.e. $x_i=\mu_i+\delta_i$ and $\mathbf{y}_i=\upsilon_i+\epsilon_i$, where δ_i (7×1) and ϵ_i (7×1) are normally distributed noise vectors. The signal to noise (S/N) ratios in the second and third subproblems are 8 and 4, respectively.

Figure 9 shows the cross-validation R^2 values for the simulations. All of the training R^2 values were close to 1. We note the following observations:

1. The multivariate NLPLS boxes are longer than the univariate NLPLS boxes and the multivariate NLPLS box plots have more extreme values. Thus, multivariate NLPLS is more sensitive to starting values. We conjecture that this is a result of the multivariate NLPLS models having more parameters and thus a larger parameter space.
2. Univariate NLPLS is more sensitive to noise. The multivariate NLPLS models give better predictions for the response variables on the high noise (SN=4) problem than univariate NLPLS.
3. The best multivariate NLPLS fits are better than univariate NLPLS, although the variance of the univariate NLPLS fits is smaller. We attribute the larger variance of the multivariate NLPLS models to their having more parameters.
4. The predictions of y_3 are worse than those of y_1 or y_2 for multivariate NLPLS. We suspect that the reason for this is that the relationship between y_3 and the predictor variables is more nonlinear than the relationship between y_1 or y_2 and the predictor variables.

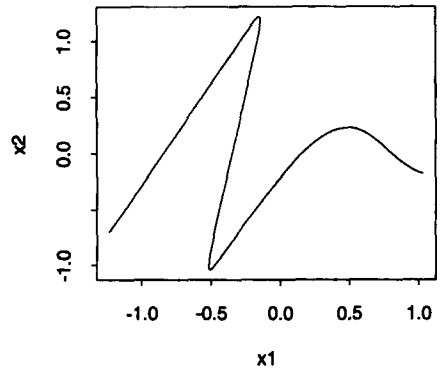

Fig. 8. A sequential NLPLS curve extracted for the first factor in a model that has small R^2 values for subsequent factors. The curve tried to wiggle throughout the entire domain and we suspect this makes it difficult for subsequent factors to model the response variable residuals.

6.3. *Composite materials*

An application of NLPLS is in the process control of the fabrication of composite materials. Composite materials are very expensive to manufacture because of (1) high scrap rates[11]; (2) rework; and (3) labor-intensive methods of inspection[12]. We investigated how information on certain quality variables like refractive index, temperature, and phase of cure cycle can be used by the process engineer to modify the process and avoid producing scrap material. Many of the quality variables are difficult or expensive to measure directly. Alternatively, the quality variables can be estimated from other process data. During fabrication, a sensor can be included in the material to collect infrared spectroscopy data, which describe the absorption properties of the materials being fabricated at different wavelengths over the curing process. We are investigating how these data can be used to predict the key-quality variables.

Northwestern University's Basic Industry Research Laboratory (BIRL) has developed and tested a sensor to collect infrared spectroscopy data and collected composite material data. They used their sensor to measure absorption properties at $p=466$ different wavelengths. Three measurements were taken at 28 different times during the cure process, giving $28\times3=84$ observations. An NLPLS model was fitted and one of the points was identified as an outlier (identified on Fig. 10). The outlier was removed and the remaining 83 observations were split into training and cross-validation sets. We selected one time at random from each of the three phases of the cure cycle and the three observations at the three times were used for cross validation giving $3\times3=9$ CV points. The remaining 74 points were used for training. Because training a neural network with 466 input nodes would be difficult, we reduced first the dimension of the predictor variables with principal components analysis. The first three principal components of the 74×466 training matrix accounted for 99.90% of the variance and the first three principal component score vectors were used as predictor variables ($p=5$) instead of the original 466 to avoid having a neural network with 466 input nodes. To visualize the data, Fig. 10 plots the first three principal component scores. The predictors

[11] Scrap rates can be as high as 40% (Fildes, 1995).

[12] Twenty-five percent of labor costs are for post-process inspection (Fildes, 1995).

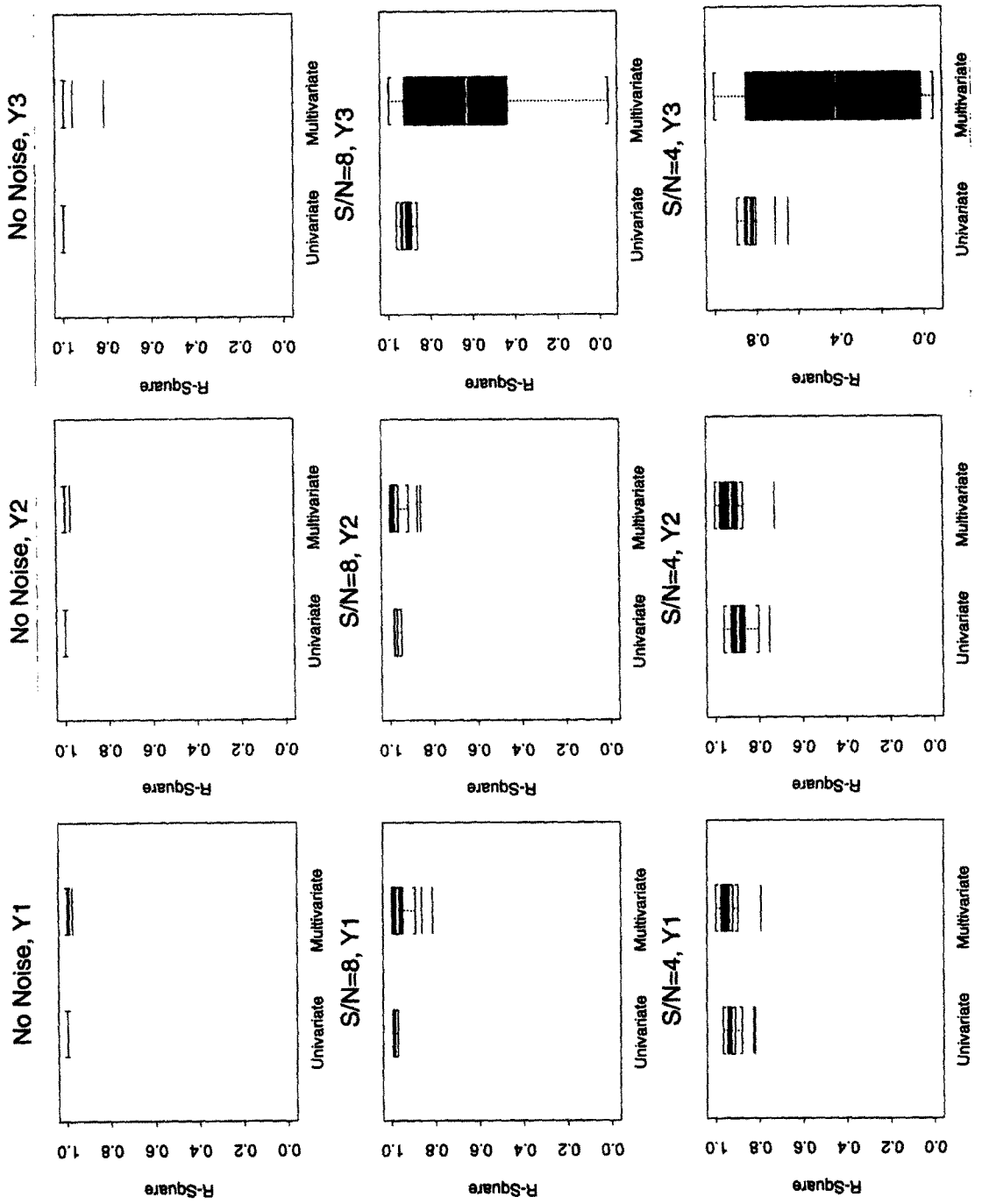

Fig. 9. Box plots of cross-validation R^2 values for surface problem with multivariate response variables using different starting values.

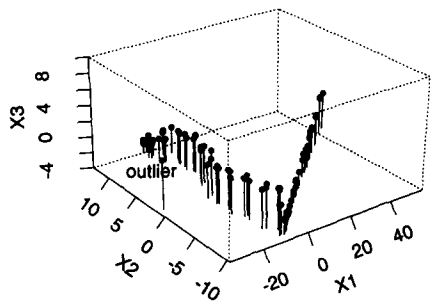

Fig. 10. Plot of predictor variables in composite-materials example.

approximately follow a V-shaped curve. NLPLS can be used to extract this curve and reduce the noise in the predictor variables and the one-dimensional scores contain nearly the same information as the original 466 coordinates.

We predicted the refractive index and temperature with NLPLS[13], three-layer neural networks (NN), PLS, PCR, and projection pursuit pursuit regression (PPR). We fitted the NLPLS and NN models ten times with different starting values. We fitted PLS, PCR, and projection pursuit models with $r=1,...,5$ and selected the model giving the largest cross-validation R^2. The box plots in Fig. 11 show the R^2 values for the training and cross-validation data. All of the methods give good predictions to the temperature response variable, but the NLPLS models are sensitive to starting values. For refractive index, NN models gave the best predictions followed by the best NLPLS models, PLS, PCR, and PPR. Projection pursuit seems to overfit the data because it has the highest R^2 value on the training data, but a smaller R^2 value on the cross-validation data. The variance of the R^2 values for NN models was very small compared with the variance for NLPLS models. The NN models were thus more robust to starting values than the NLPLS models. We attribute this to the NLPLS models having more parameters than the NN models.

The NLPCA score values can be used to determine the phase in the cure cycle. The cure cycle follows the V-shaped curve in Fig. 10. Phase 1 occurs along the left-hand side of the V, where there is large variation along the second principal axis and almost none along the first principal axis. Phase 2 occurs around the vertex of the V where there is a rapid change along the third principal axis. Phase 3 occurs along the right-hand side of the V, where there is variation along all three principal axes. Thus, score values[14] less than some constant a_1 indicate that the process is in Phase 1, score values greater than some constant a_2 indicate Phase 3, and scores between a_1 and a_2 indicate Phase 2.

[13] Simultaneous and sequential NLPLS models are identical, since only one nonlinear factor must be extracted to model the relationship between predictor and response variables.

[14] We assume that the curve has a parameterization that increases with phase.

7. Conclusions

Our conclusions regarding NLPLS are as follows:

1. When the observed predictor variables lie on a curve or surface, the simultaneous NLPLS models can produce a more parsimonious model, i.e. a smaller r and thus fewer score vectors, than projection-based methods. When the observed predictor variables do not lie in a nonlinear subspace, other nonparametric regression methods should be used, since NLPLS has no advantage over the other methods in this case.
2. Because NLPLS models the observed predictor variables with a curve or surface, outliers and points requiring extrapolation can be detected easily.
3. NLPLS has the same approximation properties as a three-layer NN.
4. Sequential NLPCA is highly sensitive to starting values and has a tendency to overfit the data. Projecting the predictor variables onto curves seems to give the model too much flexibility and our empirical results suggest that a simultaneous NLPLS model or a projection-based regression model will give better predictions on cross-validation data sets. Therefore, we do not recommend sequential NLPLS.
5. Multivariate NLPLS seems more sensitive to starting values than univariate simultaneous NLPLS. Univariate NLPLS is more sensitive to noise when there are multiple response variables that are related to each other; in these cases, we conjecture that the reason for this is that multivariate NLPLS models the relationship among the predictor variables.
6. In general, the quality of predictions measured in terms of R^2 from simultaneous NLPLS models is roughly as good as the quality of predictions from three-layer neural networks and projection pursuit. In situations favorable to NLPLS, it can provide marginally better estimates. However, the NLPLS solutions are much more sensitive to starting values and require more computational effort than three-layer neural networks or projection pursuit.

Based on our empirical experience, the projection-based regression methods give equally good prediction accuracy with less computational effort than simultaneous NLPLS. The only situation where NLPLS has a clear advantage over projection-based methods is when the predictor variables lie in a nonlinear subspace and the modeler is interested in outlier or extrapolation detection. However, we feel that many of the limitations of NLPLS can be overcome. Thisted (1988, p. 19) wrote, "[Projection pursuit] requires so much computation that even ten years ago its routine use would have been hideously expensive. That is no longer the case." As faster computers and better nonlinear optimization algorithms become available, perhaps the same will be said about NLPLS.

Acknowledgements

This work was done with computational support from NFS grant DMS-9505799. The authors thank Thomas

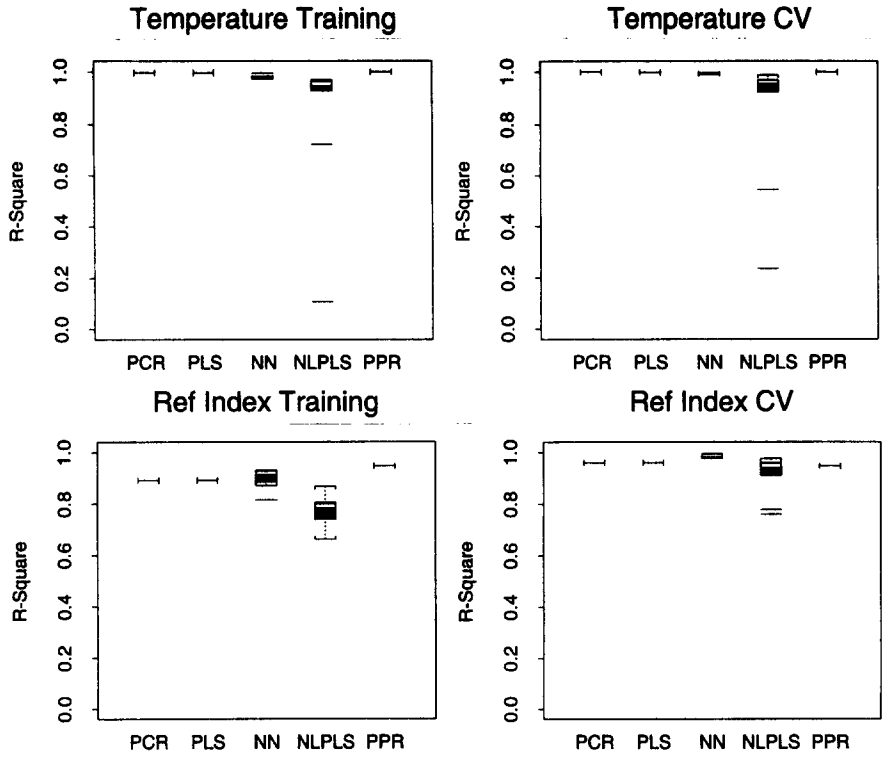

Fig. 11. The R^2 values for composite-materials problem using different starting values.

Severini for many helpful discussions, Jorge Nocedal for making his limited-memory BFGS code available, John Fildes and Northwestern University's basic industrial research laboratory for making their composite materials data available, and three referees for their constructive criticism of an earlier draft.

References

Breiman L. and J. Friedman, Predicting multivariate responses in multiple linear regression. Technical Report Technical Report Number 111, Laboratory for Computational Statistics, Department of Statistics, Stanford University. To be published in *J. R. Stat. Soc., Ser. B* (1994).

Briesch R. and E. Malthouse, Nonparametric partial least squares: bringing structural equations modeling into the nonparametric age. Working paper, Kellogg Department of Marketing, Northwestern University (1994).

Brooks, R. and Stone, M. (1994) Joint continuum regression for multiple predictands. *JASA* **89**, 428 1374–1377.

Cattell, R. (1966) The scree test for the number of factors. *Multivar. Behav. Res.* **1**, 245–276.

Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Mathl Contr. Sign. Syst.* **2**, 303–314.

Diaconis, P. and Shahshahani, M. (1984) On nonlinear functions of linear combinations. *SIAM J. Sci. Stat. Comput.* **5**, 1 175–191.

Fildes J., Personal communication (1995).

Frank I., NNPPSS: neural networks based on PCR and PLS components nonlinearized by smoothers and splines. INCINC94 Chemometrics Conference (1994).

Frank, I. and Friedman, J. (1993) A statistical view of some chemometrics regression tools. *Technometrics* **35**, 2 109–148. With discussion.

Friedman, J. and Stuetzle, W. (1981) Projection pursuit regression. *JASA* **76**, 817–823.

Garthwaite, P. H. (1994) An interpretation of partial least squares. *JASA* **89**, 425 122–127.

Horn, J. and Engstrom, R. (1979) Cattell's scree test in relation to Bartlett's chi-square test and other observations on the number of factors problem. *Multivar. Behav. Res.* **27**, 335–354.

Hwang J., S. Lay, M. Maechler, D. Martin and J. Schimert (1995) Regression modeling in back-propagation and projection pursuit learning, *IEEE Trans. Neural Networks* **5**, 342–350.

Kramer, M. (1991) Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* **37**, 2 233–243.

Liu, D. and Nocedal, J. (1989) On the limited memory BFGS method for large scale optimization. *Mathl Progr.* **45**, 503–528.

Malthouse E., *Nonlinear Partial Least Squares*. Ph.D. thesis, Northwestern University (1995).

Mardia K., J. Kent and J. Bibby, *Multivariate Analysis*. Academic Press, London (1979).

Qin, S. and McAvoy, T. (1992) Nonlinear PLS modeling using neural networks. *Comput. Chem. Engng* **16**, 4 379–391.

Thisted R., *Elements of Statistical Computing*. Chapman and Hall, New York (1988).

Wold, S. (1992) Nonlinear partial least squares modelling II. Spline inner relation. *Chemomet. Intell. Lab. Syst.* **14**, 71–84.

Wold, S., Kettaneh-Wold, N. and Skagerberg, B. (1989) Nonlinear PLS modelling. *Chemomet. Intell. Lab. Syst.* **7**, 53–65.