Nonlinear Principal Component Analysis by Neural Networks: Theory and Application to the Lorenz System

ADAM H. MONAHAN

Oceanography Unit, Department of Earth and Ocean Sciences, and Crisis Points Group, Peter Wall Institute for Advanced Studies, University of British Columbia, Vancouver, British Columbia, Canada

(Manuscript received 28 October 1998, in final form 7 May 1999)

ABSTRACT

A nonlinear generalization of principal component analysis (PCA), denoted nonlinear principal component analysis (NLPCA), is implemented in a variational framework using a five-layer autoassociative feed-forward neural network. The method is tested on a dataset sampled from the Lorenz attractor, and it is shown that the NLPCA approximations to the attractor in one and two dimensions, explaining 76% and 99.5% of the variance, respectively, are superior to the corresponding PCA approximations, which respectively explain 60% (mode 1) and 95% (modes 1 and 2) of the variance. It is found that as noise is added to the Lorenz attractor, the NLPCA approximations remain superior to the PCA approximations until the noise level is so great that the lower-dimensional nonlinear structure of the data is no longer manifest to the eye. Finally, directions for future work are presented, and a cinematographic technique to visualize the results of NLPCA is discussed.

1. Introduction

Principal component analysis (PCA), also known as empirical orthogonal function (EOF) analysis, has been widely used in oceanography and meteorology since its introduction to these fields by Lorenz (1956). PCA is an objective technique used to detect and characterize optimal lower-dimensional linear structure in a multivariate dataset, and it is one of the most important methods in the geostatistician’s multivariate statistics toolbox. Consequently, it has been well studied, and standard references exist describing the method and its implementation (Preisendorfer 1988; Wilks 1995). Its applications include reduction of data dimensionality for data interpretation (e.g., Barnston and Livezey 1987; Miller et al. 1997) and for forecasting (e.g., Barnston and Ropelewski 1992; Tangang et al. 1998). Furthermore, the connection between the the results of PCA, which are statistical in nature, and the underlying dynamics of the system under consideration are understood in some detail (North 1984; Mo and Ghil 1987).

PCA can be regarded as a single example of a general class of feature extraction methods, which attempt to characterize lower-dimensional structure in large multivariate datasets. By construction, PCA finds the lower-dimensional hyperplane that optimally characterizes the data, such that the sum of squares of orthogonal deviations of the data points from the hyperplane is minimized. If the structure of the data is inherently linear (e.g., if the underlying distribution is Gaussian), then PCA is an optimal feature extraction algorithm; however, if the data contain nonlinear lower-dimensional structure, it will not be detectable by PCA.

In the early 1990s, a neural-network-based generalization of PCA to the nonlinear feature extraction problem was introduced in the chemical engineering literature by Kramer (1991), who referred to the resulting technique as nonlinear principal component analysis (NLPCA). Another solution to this problem, coming from the statistics community, was put forward independently by Hastie and Stuetzle (1989), who named their method principal curves and surfaces (PCS). Recently, Malthouse (1998) demonstrated that NLPCA and PCS are closely related, and are, for a broad class of situations, essentially the same. Kramer’s NLPCA has been applied to problems in chemical engineering (Kramer 1991; Dong and McAvoy 1996), psychology (Fotheringham and Baddeley 1997), and image compression (De Mers and Cottrell 1993), but apart from a single unpublished report by Sengupta and Boyle (1995), we have not yet seen it applied to the large multivariate datasets common in oceanic and atmospheric sciences.

This paper is the first in a series of investigations of the application of NLPCA to the analysis of climate
data. In section 2, we introduce the general feature extraction problem and the PCA approach. We then introduce the NLPCA algorithm and show how it is in fact a natural nonlinear generalization of PCA. In section 3, we apply NLPCA to a synthetic dataset sampled from the Lorenz attractor (Lorenz 1963), with subsequent addition of noise to simulate measurement error, and show that NLPCA is better able to characterize the lower-dimensional nonlinear structure of the data than is PCA. Finally, conclusions and directions for future research are presented in section 4.

2. Feature extraction problems

We denote by \( X(t) \) a typical meteorological or oceanographic dataset, where \( i \in (1, \ldots, M) \) labels spatial stations and \( n \in (1, \ldots, N) \) labels observation times. It is usually the case that the field values at different stations do not evolve independently, that is, that the temporal variability of \( X(t) \) includes contributions from large-scale, spatially coherent features. In such a circumstance, the data will not be scattered evenly through the phase space of stations, but will tend to cluster around lower-dimensional surfaces; it is then appropriate to describe \( X(t) \) by the model

\[
X(t) = f(s_t[X(t)]) + \epsilon_n = \tilde{X}(t) + \epsilon_n.
\]

The function \( s_t : \mathbb{R}^M \to \mathbb{R}^p, 1 \leq P < M \) parameterizes a manifold of dimensionality lower than that of \( X(t) \). \( f : \mathbb{R}^p \to \mathbb{R}^M \) is a smooth map from this manifold to the original space, and the \( \epsilon_n \) are residuals. The determination of \( s_t \) and \( f \) from the data \( X(t) \), subject to an optimality criterion such as minimizing the sum of squares of the residuals, is example of a feature extraction problem: given a noisy dataset \( X(t) \), it is desired to retrieve the signal \( \tilde{X}(t) = f(s_t[X(t)]) \). Doing so, we treat the \( M \) dimensionality of \( X(t) \) as being in a sense only superficial, as the signal with which we are concerned lives on a \( P \)-dimensional submanifold of \( \mathbb{R}^M \). Because once we have found \( f \) and \( s_t \), we no longer need to work with the signal in \( \mathbb{R}^M \) and can concentrate instead on the signal in \( \mathbb{R}^p \), feature extraction can also be thought of as reduction of data dimensionality.

The method of feature extraction most common in the atmospheric and oceanic sciences is PCA, which optimally extracts linear features from the data. However, if the underlying structure of the data is nonlinear, traditional PCA is suboptimal in characterizing this lower-dimensional structure. This motivates the definition of NLPCA. In this section, we provide a brief review of PCA and demonstrate how it generalizes naturally to a nonlinear method. We then discuss the NLPCA method in detail.

a. Principal component analysis

Traditional PCA can be formulated as a special case of the feature-extraction problem, in which the data \( X(t) \) (assumed, without loss of generality, to have zero mean in time) is fit to the linear \( P \)-dimensional model

\[
X(t) = \sum_{k=1}^{P} [X(t_k) \cdot e_k] e_k + \epsilon_n,
\]

for vectors \( e_k \in \mathbb{R}^M \), such that the sum of squares of the residuals,

\[
J = \langle [X - \tilde{X}]^2 \rangle,
\]

is a minimum, where angle brackets denote a sample time average. The vector \( e_k \) is the \( k \)th EOF and the projection of \( X(t) \) on \( e_k \) is the \( k \)th principal component (PC). The product of the \( k \)th EOF with the \( k \)th PC defines a vector time series usually referred to as the \( k \)th PC mode. The PCA approximation \( \tilde{X}(t) \) to \( X(t) \) lives on the \( P \)-dimensional hyperplane that passes optimally through the middle of the data (von Storch and Zwiers 1999). PCA has the variance partitioning property

\[
\sum_{i=1}^{M} \text{var}(X_i) = \sum_{i=1}^{M} \text{var}(\tilde{X}_i) + \sum_{i=1}^{M} \text{var}(X_i - \tilde{X}_i),
\]

so it is sensible to say that \( \tilde{X}(t) \) “explains” a certain fraction of the variance of \( X(t) \). In particular, \( \tilde{X}(t) = [X(t) \cdot e_1] e_1 \) is the one-dimensional linear approximation to \( X(t) \), which explains the highest percentage of the variance. The fraction of variance explained by \( \tilde{X}(t) \) is a nondecreasing function of the approximation dimension \( P \); increasing the dimensionality of the PCA approximation increases its fidelity to the original data.

PCA is usually thought of in terms of the eigenstructure of the data covariance matrix \( C = (X^T X)^{-1} \). In fact, the vectors \( e_k \) are eigenvectors of \( C \) corresponding to the \( P \) largest eigenvalues,

\[
Ce_k = \lambda_k e_k,
\]

where \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_P \). This fact follows from the minimization of (3) subject to the constraint that the vectors \( e_k \) are normalized. While diagonalization of \( C \) is the standard approach to calculating the \( e_k \), it has no analog in the nonlinear generalization to be considered. We concentrate on the variational formulation of PCA (from which follows its relation to eigenanalysis of the covariance matrix) because it generalizes naturally to the nonlinear feature extraction problem.

We note that the PCA approximation \( \tilde{X}(t) \) to \( X(t) \) is the composition of two functions:

1. a projection function \( s^P : \mathbb{R}^M \to \mathbb{R}^P \),

\[
s^[(P)]=X(t)\cdot e_i, \ldots, X(t)\cdot e_p]^T = \Pi X(t).
\]

2. an expansion function \( f^P : \mathbb{R}^P \to \mathbb{R}^M \),

\[
f^[(P)](s^[(P)]) = \Pi s^[(P)].
\]

Thus, the PCA approximation \( \tilde{X}(t) \) to \( X(t) \) is given by

\[
\tilde{X}(t) = (f \circ s)[X(t)] = \Pi^T[\Pi X(t)].
\]
rons can approximate to arbitrary accuracy any continuous function from $\mathbb{R}^s$ to $\mathbb{R}^r$, if the number of neurons in the second layer is sufficiently large. Feed-forward neural networks are described in detail in Bishop (1995), and their application to meteorology and oceanography is reviewed in Hsieh and Tang (1998).

Another solution to the general feature extraction problem was introduced independently by Hastie and Stuetzle (1989). Their method, termed PCS, is based on a rather different set of ideas than NLPCA. In practice, however, because both minimize the sum of squared errors (10), the two methods both boil down to iterative solutions of a nonlinear variational problem. In fact, Malthouse (1998) argued that NLPCA and PCS are quite similar for a broad class of feature extraction problems. A primary difference between NLPCA and PCS is that in the former the projection function $s_i$ is constrained to be continuous, while in the latter it may have a finite number of discontinuities. Although in this paper we will investigate the use of Kramer’s NLPCA, because its implementation is straightforward, PCS has a stronger theoretical underpinning, and we will use the connection between the two methods noted by Malthouse to make hypotheses about certain properties of NLPCA.

Figure 1 shows the architecture of the five-layer network used to extract the 1D NLPCA approximation to the data set $X(t_n) \in \mathbb{R}^M$; this network is unusual in that the third layer contains only a single neuron. We shall refer to this third layer as the bottleneck layer. The first (input) and fifth (output) layers each contain $M$ neurons. Layers 2 and 4 are called, respectively, the encoding and decoding layers; they contain $L$ neurons, the transfer functions of which are hyperbolic tangents. The transfer functions of the bottleneck and output layers are linear.

As input, the network is presented the vector $X(t_n)$ for each time $t_n$; the corresponding network output is denoted $\hat{X}[X(t_n)]$. The weights and biases are adjusted (“trained”), using a conjugate gradient algorithm (Press et al. 1992) until the sum of squared differences between input and output,
\[ J = \langle \| \mathbf{X}(t_n) - \mathbf{X}(t_n) \| \rangle, \quad (12) \]
is minimized (subject to certain caveats discussed in the next section). As the network is trained to approximate as closely as possible the input data itself, it is said to be \textit{autoassociative}. It was proved by Sanger (1989) that if the transfer functions of the neurons in the second and fourth layers are linear, the resulting network performs classic PCA, such that the output of the bottleneck layer is the time series \( s(\mathbf{t}_n) \) of Eq. (6) (up to a normalization factor).

We now consider the manner by which this network solves the feature extraction problem for \( P = 1 \). The first three layers, considered alone, form a map from \( \mathbb{R}^M \) to \( \mathbb{R} \), and the last three layers alone form a map from \( \mathbb{R} \) to \( \mathbb{R}^M \). All five layers together are simply the composition of these two maps. Because the bottleneck layer contains only a single neuron, the network must compress the input down to a single one-dimensional time series before it produces its final \( M \)-dimensional output. Once the network has been trained, the output \( \mathbf{X}(t_n) \) is the optimal one-dimensional approximation to \( \mathbf{X}(t_n) \), embedded in \( \mathbb{R}^M \). Using Cybenko’s result, if \( L \) is sufficiently large, then the first three layers can approximate any continuous \( s_1 \), and the last three layers any continuous \( f \), to arbitrary accuracy. Thus, the network illustrated in Fig. 1 should be able to recover optimally, in a least squares sense, any one-dimensional nonlinear structure present in \( \mathbf{X}(t_n) \). It is not required that the encoding and decoding layers each have the same number of neurons, but we fix the numbers to be the same so as to have only one free parameter in the model architecture. That the network must be composed of (at least) five layers follows from the fact that each of the functions \( s_1 \) and \( f \) requires a network with (at least) three layers for its approximation. The composition \( f \circ s_1 \) of the two must then have at least five layers, as one layer is shared.

As noted by LeBlanc and Tibshirani (1994), Hastie and Stuetzle’s PCS partitions variance in the same fashion as does traditional PCA: if \( \mathbf{X}(t_n) \) is the PCS approximation to \( \mathbf{X}(t_n) \), then

---

**Fig. 2.** The Lorenz attractor, projected on the \( x_1 \), \( x_2 \), \( x_3 \), and \( x_1 \) planes.
As with PCA, it is therefore sensible to describe a PCS approximation as explaining a certain fraction of variance in the original dataset. From the close relationship between NLPCA and PCS demonstrated by Malthouse (1998), it is tempting to hypothesize that NLPCA also partitions variance in such a fashion. While we are not aware of a rigorous proof of this result, this partitioning of variance occurs in all of the examples we have considered, and in the following discussion we shall assume that Eq. (13) holds for $\mathbf{X}(\mathbf{t}_n)$ the NLPCA approximation to $\mathbf{X}(\mathbf{t}_n)$.

The network illustrated in Fig. 1 will extract the optimal one-dimensional curve characterizing $\mathbf{X}(\mathbf{t}_n)$. To uncover higher-dimensional structure, the number of neurons in the bottleneck layer can be increased. For example, if two neurons are used, the network will determine the optimal two-dimensional characterization (by continuous functions) of $\mathbf{X}(\mathbf{t}_n)$. In general, a $P$-dimensional NLPCA approximation to $\mathbf{X}(\mathbf{t}_n)$ can be obtained by setting to $P$ the number of neurons in the bottleneck layer.

Because the traditional PCA model has the additive structure (2), the optimal linear $P$-dimensional linear substructure of $\mathbf{X}(\mathbf{t}_n)$ can be found all at once, or mode by mode; both methods yield the same result. In the latter approach, the first mode $\mathbf{X}^{(1)}(\mathbf{t}_n)$ of $\mathbf{X}(\mathbf{t}_n)$ is calculated from the entire dataset, and then the second mode is calculated from the residual $\mathbf{X}(\mathbf{t}_n) - \mathbf{X}^{(1)}(\mathbf{t}_n)$, taking advantage of the fact that the second PC mode of $\mathbf{X}(\mathbf{t}_n)$ is the first PC mode of this residual. The two approaches are equivalent for PCA because the most general linear function of $P$ variables has the additive structure

$$g(u_1, u_2, \ldots, u_P) = \sum_{i=1}^{P} \alpha_i u_i.$$ (14)

They are generally distinct, however, for NLPCA, as an
arbitrary smooth function \( f \) of \( P \) variables cannot be decomposed as a sum of smooth functions of one variable. That is, in general,

\[
f(u_1, u_2, \ldots, u_p) \neq f_1(u_1) + f_2(u_2) + \cdots + f_p(u_p)
\]

for some functions \( f_1, f_2, \ldots, f_p \); we cannot usually write \( f \) as a generalized additive model (Hastie and Tibshirani 1990). There are thus a number of different ways to obtain a \( P \)-dimensional NLPCA representation of the data, from “all at once” to “mode by mode,” and including all variations in between. These approaches are not equivalent. The one that is appropriate will presumably depend on the problem being considered. In principle, the all-at-once method will produce the \( P \)-dimensional approximation that explains the highest fraction of the variance and thus is the optimal approach, in principle, the all-at-once method will produce the \( P \)-dimensional approximation that explains the highest fraction of the variance and thus is the optimal approach, although it is conceivable that the mode-by-mode approach might be appropriate in certain circumstances. These approaches are not equivalent. The one that is appropriate will presumably depend on the problem being considered. In principle, the all-at-once method will produce the \( P \)-dimensional approximation that explains the highest fraction of the variance and thus is the optimal approach.

In this paper, the phrase “P-dimensional NLPCA” will refer to the results of the all-at-once approach.

c. Implementation

Neural networks are powerful tools for function approximation. Given input and target datasets \( u(t_n) \) and \( v(t_n) \), \( n = 1, \ldots, N \), a neural network, denoted by \( \mathcal{N} \) can be trained until \( \mathcal{N}[u(t_n)] \) is an arbitrarily good approximation to \( v(t_n) \), given a sufficiently large number of neurons. That is, a network can always be built so that the total sum of squared errors,

\[
\langle \| v(t_n) - \mathcal{N}[u(t_n)] \|^2 \rangle,
\]

is as small as desired. Another important property of the neural network is that it generalizes, that is, that given new data \( u(t_{N+1}), v(t_{N+1}) \), the network error on this data is about the same size as the errors over the training set:

\[
\| v(t_{N+1}) - \mathcal{N}[u(t_{N+1})] \| \approx \| v(t_n) - \mathcal{N}[u(t_n)] \|. \tag{17}
\]

The two goals of minimizing network error and maximizing its ability to generalize are often incompatible, and a subtle balance must be struck between the two. This situation arises, for example, in the case when \( u(t_n) \) and \( v(t_n) \) are of the form

\[
u(t_n) = z(t_n) + \epsilon_n,
\]

\[
v(t_n) = f(z(t_n)) + \eta_n,
\]

where \( \epsilon_n \) and \( \eta_n \) are noise terms, and it is desired that \( \mathcal{N} \) learn the deterministic relationship \( f \) between \( u(t_n) \) and \( v(t_n) \). In such a case, care must be taken to avoid allowing the network to fit the noise as well. If \( \mathcal{N} \) is trained until it maps particular details of a given realization of \( u(t_n) \) into those of a given realization of \( v(t_n) \), and thus will not generalize, the network has overfit. An overfit network is not truly representative of the structure underlying a dataset.

Developing methods for the avoidance of overfitting is a major area of research in the field of neural networks (Finnoff et al. 1993). Of these, an important subclass is made up of early stopping techniques, in which the network training is stopped before the error function is minimized, according to some well-defined stopping criterion. The essential idea behind the early stopping is that the network is trained for a number of iterations sufficient to model any true underlying structure, but insufficient to fit the noise exactly. While the benefit of stopping network training before the error function is minimized may seem counterintuitive at first, experiments have demonstrated that early stopping algorithms improve the resulting neural network’s ability to generalize (Finnoff et al. 1993). The strategy we used was to set aside a fraction \( \alpha \) (typically about 30%) of the data points, selected randomly, into a validation set not used to train the network. As network training proceeded, the network error on the validation set was monitored, and training was stopped when this validation set error started to increase, or after 500 iterations, whichever was the first to occur. Again, the validation set was not used to train the network, but only to monitor its ability to generalize. The degradation of this generalization ability was the stopping criterion employed.

Once training stopped, the fraction of unexplained variance (FUV) over the entire dataset was calculated. Because NLPCA was found experimentally to partition variance as described by Eq. (13), the FUV is simply the variance of the target dataset less that of the model output, divided by the variance of the data:

\[
\text{FUV} = \sum_{i=1}^{M} \text{var}(X_i) - \sum_{i=1}^{M} \text{var}(\hat{X}_i) \over \sum_{i=1}^{M} \text{var}(X_i). \tag{20}
\]

By design, NLPCA should minimize FUV (subject to the constraints discussed above). However, experience indicates that the network with the smallest value of FUV may possess other undesirable features. In particular, the NLPCA approximation may lack symmetries manifest in the original dataset, or, instead of being smooth, the NLPCA approximation may be twisted and coiled. In both cases, the NLPCA approximation also tended to be irreproducible between independent network trainings and thus did not robustly characterize the data. It was decided that the NLPCA approximation should also reflect obvious symmetries in the original data, should be smooth, and should be reproducible, even if the FUV was not quite the minimum obtainable. The first two criteria were simply gauged by inspection of the NLPCA approximation; the lack of objective measures for these is admittedly unsatisfactory. The possibility of including a spline-type regularization term in the error function (12) to enforce smoothness is one of the options being considered for future work. The third criterion, reproducibility of results, has a natural objec-
tive measure: a quantitative measure of the difference between NLPCA approximations $\mathbf{X}^{(1)}(t_n)$ and $\mathbf{X}^{(2)}(t_n)$ to the dataset $\mathbf{X}(t_n)$ is the normalized mean square distance (NMSD), defined as

$$\text{NMSD} = \frac{\langle \|\hat{\mathbf{X}}^{(1)} - \hat{\mathbf{X}}^{(2)}\|^2 \rangle}{\sum_i \text{var}(X_i)}.$$  \hspace{1cm} (21)

Experiments with this statistic indicate that two approximations are essentially indistinguishable if their NMSD is about 2% or less. For NMSD below 20%, the approximations have the same gross features but may differ in details. When the NMSD between two NLPCA approximations exceeds about 40%, they differ considerably. These numbers are rough estimates determined by experimentation. While we have found the NMSD to be a useful statistic characterizing the distance between NLPCA approximations, its utility is limited by the lack of a theoretical understanding of its sampling distribution. With a more rigorous understanding of the distribution of NMSD, we could address issues such as the statistical significance of differences between curves and carry out formal hypothesis testing. Still, even in the absence of such a formal theory, the NMSD can serve as a useful quantitative, if somewhat subjective, measure of the distance between NLPCA approximations to a dataset.

The number of neurons $L$ in the encoding and decoding layers determines the class of functions that $s_i$ and $f$ can take. As $L$ increases, there is more flexibility in the forms of $s_i$ and $f$, but the model also has more parameters, implying both that the error surface becomes more complicated and that the parameters are less constrained by data. In the end, the number of neurons $L$ in the encoding and decoding layers was set to be the largest number such that the resulting NLPCA approximations conformed to the smoothness and robustness criteria discussed in the previous paragraph.

Clearly, a $P$-dimensional NLPCA model will contain more parameters than a $P$-dimensional PCA model. In fact, this is what allows NLPCA to solve more general feature extraction problems than PCA. Generally, the number of parameters in a statistical model is minimized to reduce overfitting of the data. However, this issue is addressed with the early stopping technique described above. The use of such a technique allows the number of parameters in the statistical model to be increased without substantially increasing the risk of the model overfitting the data.

Thus, the NLPCA approximation to a dataset $\mathbf{X}(t_n)$ is defined operationally as the output of the five-layer autoassociative neural network that minimizes the error (12) subject to the constraint that the network generalizes, and that the output must be smooth, reproducible, and reflect symmetries manifest in $\mathbf{X}(t_n)$. This operational definition will undoubtedly be refined as experience is gained in the implementation of NLPCA.

3. The Lorenz attractor

To investigate the implementation of NLPCA, we consider a synthetic dataset consisting of a set of points sampled from the Lorenz attractor (Lorenz 1963). This familiar object is the attractor on which (as $t \to \infty$) live solutions $\mathbf{x}(t)$ of the system of coupled nonlinear ODEs

$$\begin{align*}
\dot{x}_1 &= -\alpha x_1 + \sigma x_2, \\
\dot{x}_2 &= -x_1 x_3 + r x_1 - x_2, \\
\dot{x}_3 &= x_1 x_2 - b x_3,
\end{align*}$$

(22)–(24)

with parameter values $r = 28$, $b = 8/3$, and $\sigma = 10$. We chose to use synthetic data to test the NLPCA method because

- by adding random noise to the signal $\mathbf{x}(t)$, the sensitivity of the method to noise level can be tested, and
- the structure of the Lorenz attractor is well known and of sufficiently low dimension that visualization of results is straightforward.

Figure 2 displays the projections of the Lorenz attractor [as determined by numerical integration of Eqs. (22)–(24)] on the $x_1 x_2$, $x_2 x_3$, and $x_1 x_3$ planes. It turns out that the Lorenz attractor is fractal, with a box-counting dimension of about 2.04 (Berliner 1992). However, inspection of the butterfly-shaped attractor indicates that a one-dimensional U-shaped curve passing through the centers of the two lobes should explain a substantial fraction of the variance.

To produce a dataset of size similar to that typically encountered in climate applications (e.g., 600 points in length, corresponding to 50 yr of monthly data), the data displayed in Fig. 2 were subsampled at even intervals in time to produce a 3D time series 584 points in length, to be denoted $\mathbf{z}(t_n)$. The subsampled dataset is displayed in Fig. 3. Clearly, it has the gross structure of the original attractor. To investigate the effects of noise on the NLPCA results, we constructed the datasets

$$\mathbf{x}(t_n) = \mathbf{z}(t_n) + \eta(t_n),$$

(25)

where $\eta(t_n)$ is a 584-point 3D series of Gaussian iid random deviates with zero mean and unit standard deviation, and $\eta$ is a tunable parameter for the noise level. This noise is added in an effort to model measurement error; the stochasticity is not intrinsic to the dynamics.

The 1D PCA approximation to $\mathbf{x}(t_n)$ when $\eta = 0$ is displayed in Fig. 4; it is a straight line passing through the centers of the two lobes of the attractor and explains 60% of the variance of $\mathbf{x}(t_n)$. Figure 5 displays the 1D NLPCA approximation to $\mathbf{x}(t_n)$. As anticipated, it is a U-shaped curve passing through the middle of the data. This curve explains 76% of the variance, an improvement of 16% over the PCA results. Clearly, the 1D NLPCA approximation is substantially closer to $\mathbf{x}(t_n)$ than is the 1D PCA approximation. The network used to perform the NLPCA had three input and output neurons for $x_1$, $x_2$, and $x_3$; one bottleneck neuron; and $L =$
3 neurons in the encoding and decoding layers. Experimentation indicated that the NLPCA results improved using $L = 3$ over using $L = 2$ (i.e., the former had a smaller FUV than the latter), but that for $L > 3$, the results did not improve. Turning now to the issue of robustness of results, the NMSD between six different 1D NLPCA curves (not shown) varies between 0.5% and 2%. These curves differ only in small details and agree in their essential structure with the curve shown in Fig. 5. Thus, the 1D NLPCA approximation to $x(t_n)$ displayed in Fig. 5 is a robust result that improves substantially over the 1D PCA approximation.

Figure 6 displays the 2D PCA approximation of the data $x(t_n)$ when $\eta = 0$; this surface explains 95% of the variance. The 2D PCA approximation is a flat sheet that characterizes well the structure of the data as projected in the $x_1x_3$ and $x_2x_3$ planes but fails to reproduce the structure seen in the projection on the $x_1x_2$ plane. In Fig. 7, the result of a 2D NLPCA of $x(t_n)$ is shown. This surface explains 99.5% of the variance, implying an order of magnitude reduction in FUV as compared to the PCA result. The network used to perform the NLPCA had two neurons in the bottleneck layer and $L = 6$ neurons in the encoding and decoding layers. It was found that decreasing $L$ below 6 also decreased the fraction of variance explained, and increasing it above $L = 6$ had little effect upon the results. The 2D NLPCA result is highly robust: a sample of four NLPCA models (not shown) had NMSD between curves of at most 0.1%. As with the 1D example considered above, the NLPCA approximation is a substantially better approximation to the original dataset than is the PCA approximation.

We consider now a dataset $x(t_n)$ obtained from (25) with $\eta = 2.0$. The 1D PCA approximation to $x(t_n)$ (not shown) explains 59% of the variance. The 1D NLPCA approximation (Fig. 8), explains 74% of the variance. The curve in Fig. 8 is very similar to that shown in Fig. 5 for the $\eta = 0$ case; the two-lobed structure of the data is still manifest at a noise level of $\eta = 2.0$, and the NLPCA is able to recover it. Addressing again the issue of robustness of results, six different NLPCA approx-
Fig. 5. As in Fig. 4, for 1D NLPCA approximation.

Approximations to \( x(t_n) \) were found to have NMSD varying from 0.5% to 3%. These six curves agree in their essential details, although the set displays more variability between members than did the corresponding set for \( \eta = 0 \). Figure 9 shows the results of a 2D NLPCA performed on this dataset. This explains 97.4% of the variance, in contrast to the PCA approximation, which explains 94.2%. Thus, the FUV of the 2D NLPCA approximation is about half that of the 2D PCA approximation. These results too are robust; the NMSD between different 2D NLPCA models was about 0.2%. The 2D NLPCA approximation is again an improvement over the 2D PCA approximation, but not by such a substantial margin as was the case when \( \eta = 0 \). This is because the noise-free Lorenz attractor is very nearly two-dimensional, so the 2D NLPCA was able to account for almost all of the variance. The addition of noise acted to smear out this fine fractal structure and made the data cloud more three-dimensional. The 2D NLPCA applied to this quasi-3D structure could not produce as close an analog as was the case when \( \eta = 0 \).

At a noise strength of \( \eta = 5.0 \), the dataset \( x(t_n) \) still has a discernible two-lobed structure, but it is substantially obscured. The 1D PCA approximation (not shown) explains 54% of the variance, whereas the 1D NLPCA approximation (shown in Fig. 10) explains 65%. Again, the 1D NLPCA approximation to \( x(t_n) \) is qualitatively similar to that obtained in the noise-free case (Fig. 5), especially in the projection onto the \( x_1x_3 \) plane. The \( \eta = 0 \) and \( \eta = 5.0 \) 1D NLPCA approximations differ at the ends of the curves. Presumably, the structure represented in the former is somewhat washed out by noise in the latter. Four different NLPCA curves for the data obtained with \( \eta = 5.0 \) share their gross features but differ fairly substantially in detail. In this case, the NMSD between curves varies between 5% and 10%. The 2D NLPCA approximation to this data (not shown) explains 90% of the variance, only slightly more than the 2D PCA approximation, which explains 88% of the variance.

Finally, at a noise level of \( \eta = 10.0 \) the two-lobed structure of \( x(t_n) \) is no longer obvious, and the data cloud appears as a fairly homogeneous, vaguely ellipsoidal blob. The results of NLPCA by this noise level are no longer robust, tending to be asymmetric, convoluted curves. At this noise level, then, NLPCA may no longer
be a useful technique for characterizing low-dimensional nonlinear structure of the dataset, precisely because the addition of noise has destroyed this structure. This illustrates the point mentioned in section 2c that NLPCA is a useful, robust tool for reduction of data dimensionality when nonlinearities in the dataset are manifest, but it is perhaps untrustworthy when the data clouds possess no such obvious lower-dimensional structure.

4. Discussion and conclusions

In this paper, we have considered a nonlinear generalization of traditional principal component analysis (PCA), known as nonlinear principal component analysis (NLPCA). The generalization was carried out by casting PCA in a variational formulation as the optimal solution to the feature extraction problem posed in section 2 for which the functions \( s \) and \( f \) are constrained to be linear. When PCA is regarded in this way, natural generalizations of the method to the nonlinear case are evident. One possible approach is to allow the map \( s \) to be nonlinear while \( f \) remains linear. Such a method was considered by Oja and Karhunen (1993) and by Oja (1997), who showed that such a generalization could be carried out by a two-layer recursive neural network. Another approach is to allow both functions \( s \) and \( f \) to be nonlinear, as was suggested by Kramer (1991), who also provided an implementation using a five-layer neural network. Kramer denoted his method nonlinear PCA, as did Oja and Karhunen. It is important to note that these two methods are in fact quite different, as the expansion map used by Oja and Karhunen is still a linear function.

Another approach to the nonlinear feature extraction problem was considered by Hastie and Stuetzle (1989), who introduced principal curves and surfaces (PCS) to find lower-dimensional structures that pass optimally through the middle of data. Like Kramer’s NLPCA, PCS models are obtained as iterative solutions to nonlinear
variational problems. Kramer’s NLPCA has the advantage that it is implemented using feed-forward neural networks, for which computer packages are widely available. The advantage of PCS is that it rests on a stronger theoretical base than does NLPCA. However, as was argued by Malthouse (1998), NLPCA and PCS are in fact essentially the same for a broad class of feature extraction problems. We can thus use theoretical results proven rigorously for PCS and apply them by analogy to NLPCA. In particular, PCS can be proven to partition variance as described in Eq. (13), and we find experimentally that NLPCA does as well. It is important to remember that the two methods are not exactly identical and to be on guard for those cases in which the two methods diverge. Malthouse (1998) pointed out that because Kramer’s NLPCA is unable to model discontinuous projection and expansion functions, PCS and NLPCA will differ in those cases for which such maps are necessary. We hypothesize that the use of a seven-layer neural network to perform NLPCA, such that the projection and expansion networks can well approximate discontinuous maps, will yield an NLPCA that is in closer correspondence to PCS. This is a current subject of research and will be discussed in a future publication.

As a first experiment with NLPCA, we analyzed a 584-point dataset sampled from the Lorenz attractor (Lorenz 1963). We found that the 1D and 2D NLPCA approximations to this dataset explain 74% and 99.5% of the variance, respectively, substantially improving over the 1D and 2D PCA approximations, which explain respectively 60% and 95% of the variance. Adding noise to the Lorenz data, it was found that as long as the noise was sufficiently weak that the two-lobed structure of the Lorenz attractor was not obscured, both the 1D and 2D NLPCA approximations to the data were superior to the corresponding PCA approximations. However, when the noise was of sufficient strength that the nonlinear structure underlying the dataset was no longer manifest, the results of the NLPCA were not robust, and NLPCA ceased to be a useful method for characterizing the low-dimensional structure of the data.
PCA is useful for data exploration because its results are easily visualized. For each PCA mode, one obtains a vector $e_k$ characterizing a direction in station space and a time series $a_k(t_n) = X(t_n) \cdot e_k$ describing the projection of the data on this vector. In standard climate applications, the vector $e_k$ is presented as a map, and $a_k(t_n)$ describes the extent to which this map contributes to the data at time $t_n$. The $P$-dimensional PCA approximation to $X(t_n)$ is the superposition of the first $P$ modes:

$$\hat{X}(t_n) = \sum_{k=1}^{P} a_k(t_n) e_k. \quad (26)$$

Because of the additivity of the modes in the approximation (26), each mode can be understood individually. The time series and spatial pattern of each mode can be considered independently because of the separation of space and time assumed in the PCA expansion. The form of the PCA expansion is reminiscent of that of the solution to a linear partial differential equation by the method of separation of variables. It was through this similarity that North (North 1984) was able to make the connection between the statistics and dynamics of a certain class of linear dynamical systems.

However, the general NLPCA model (1) is neither separable nor additive. How, then, do we visualize the results? For each time $t_n$, the NLPCA provides

1) $P$ values from the bottleneck layer, and
2) a point in $\mathbb{R}^M$ corresponding to a spatial map.

A means of visualizing the results of the NLPCA, then, is through a sequence of maps running through the set $\{t_n\}_{n=0}^N$. This sequence can be thought of as a movie describing how the optimal $P$-dimensional approximation to the original data evolves through time. Such a cinematographic idea underlies the interpretation of the traditional PCA model as well, in which case the movie is of the vibrations of an oddly shaped drumhead, which can be decomposed as the superposition of a number of independently evolving modes of oscillation. NLPCA
simply allows for more complicated movies than does traditional PCA analysis. The processing speed of modern computers is sufficiently large as to make NLPCA visualization via a movie practical. Such films may prove to be useful tools in exploratory data analysis. We are at present working on such analyses of climatic datasets, in particular those associated with the ENSO phenomenon. Results from these analyses will be presented in future publications.

PCA has also been widely used as a data dimensionality reduction tool in the area of statistical forecasting. When building a statistical forecast model employing canonical correlation analysis (Barnston and Ropelewski 1992) or a neural network (Tangang et al. 1998) over a large number of grid sites, the time series of which are composed of signal plus noise, it is better to carry out the forecasts in a subspace of smaller dimension, within which a substantial fraction of the variance lives. In such cases, the forecast models predict the expansion coefficients $a_k(t_n)$ of several of the leading PCA modes of the data (the precise number of modes used is, of course, problem dependent). The forecast field is then obtained through the expansion map. It is typical in these models that the time series of only the first few modes can be forecast with appreciable skill. In general, a $P$-dimensional NLPCA approximation to a dataset explains a higher fraction of variance than does the $P$-dimensional PCA approximation. If the NLPCA time series can be equally well forecast as the PCA time series, then the reconstructed field obtained through the NLPCA expansion map will be a better forecast than the corresponding PCA reconstructed field. Thus, the use of NLPCA rather than PCA as a data dimensionality reduction tool may improve the field forecast skill of statistical forecast models. This is being investigated at present.

While NLPCA and PCS have been successfully applied to problems in a number of disciplines (Hastie and Stuetzle 1989; Kramer 1991; Banfield and Raftery 1992; De Mers and Cottrell 1993; Dong and McAvoy 1996;
Fotheringham and Baddeley 1997), the only application of NLPCA to climate data that we have been able to find is an unpublished report by Sengupta and Boyle (1995), who performed an analysis of monthly averaged precipitation fields over the contiguous United States for the years from 1979 to 1988. Their results failed to demonstrate that the results of NLPCA improved substantially over those obtained by PCA. In particular, they used a visualization technique based on correlating the data with the output of their single bottleneck neuron that suboptimally characterized the information yielded by the NLPC analysis. As discussed above, the results of an NLPC analysis of gridded data have a natural cinematographic interpretation not captured by such a simple correlation. The results presented above for the Lorenz data, and the preliminary results of an NLPC analysis of tropical Pacific sea surface temperature (to be presented in a future publication), indicate that NLPCA is a more powerful tool for the reduction of data dimensionality than PCA.

Acknowledgments. The author would like to acknowledge William Hsieh, Benyang Tang, and Lionel Pandolfo for helpful advice during the course of this work, and an anonymous referee whose comments brought to the author’s attention one or two points that were unclear in the original draft. The author would like to acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada via grants to W. Hsieh, and the Peter Wall Institute for Advanced Studies.

REFERENCES

