Some theoretical results on nonlinear principal components analysis

Conference Paper · July 1995

DOI: 10.1109/ACC.1995.529349 · Source: IEEE Xplore

Edward C Malthouse
Northwestern University
205 PUBLICATIONS 7,916 CITATIONS

Some Theoretical Results on Nonlinear Principal Components Analysis

E.C. Malthouse, R.S.H. Mah, and A.C. Tamhane [1]
Departments of Statistics and Chemical Engineering

Northwestern University
Evanston, IL 60201
ecm@galton.stats.nwu.edu

Abstract

We present some results on the NonLinear Principal Components Analysis (NLPCA) method of doing nonlinear feature extraction and discuss its relation to the principal curve/surface method. Both methods attempt to reduce the dimension of a set of multivariate observations by fitting a curve or surface through the middle of the observations and projecting the observations onto this curve/surface. The two methods fit their models under a similar objective function, with one important difference: NLPCA defines the function which maps observed variables to scores (projection index) to be continuous. We show that the effects of this constraint are (1) NLPCA is unable to model curves and surfaces which intersect themselves and (2) the NLPCA "projections" are suboptimal producing larger approximation error. We show how NLPCA score values can be interpreted and give the results of a small simulation study comparing the two methods.

1. Introduction

Suppose that an $n \times p$ matrix $\mathbf{X}$ contains n observations (subjects, cases, items, etc.) on p variables (process variables, attributes, etc.)[1]. The superficial dimension of the observations is p, but the p measurements often contain similar information and the intrinsic dimension (call it r) of the observations is much smaller than p. It is often useful to extract a new set of variables called *features* which contain the same information as $\mathbf{X}$, but have the smaller intrinsic dimension[2]. We will call the values of these feature variables *scores* and denote them by $n \times r$ matrix $\mathbf{S}$. We hypothesize that the observations and scores are related as follows:

$$\mathbf{x} = \mathbf{f}(\mathbf{s}) + \epsilon = \begin{pmatrix} f_1(\mathbf{s}) + \epsilon_{\mathbf{1}} \\ \vdots \\ f_p(\mathbf{s}) + \epsilon_{\mathbf{p}} \end{pmatrix}, \qquad (1)$$

where $\mathbf{x}$ is a $p\times 1$ observation, ϵ is a vector of noise, $\mathbf{s}$ is an $r \times 1$ row vector of scores, and $\mathbf{f}$ is a smooth function which is called (globally) parameterized r-surface in $\Re^p$. When $\mathbf{s}$ is unidimensional ($r = 1$), surface $\mathbf{f}$ is usually called a *curve*. We give some examples of curves in Section 3. The feature extraction problem is to find $\mathbf{f}$ and $\mathbf{S}$.

Several solutions have been proposed to the feature extraction problem. Principal components analysis (PCA) [2] provides one solution to this problem. PCA reduces the data to its intrinsic dimension by finding an r-dimensional hyperplane (spanned by the columns of $p \times r$ matrix $\mathbf{U}$) which satisfies the following minimum distance property:

$$\mathbf{U} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \operatorname{proj}_{\mathbf{A}}\mathbf{X}\|^{\mathbf{2}}, \qquad (2)$$

where $\mathbf{A}$ is a $p \times r$ matrix and $\operatorname{proj}_{\mathbf{A}}\mathbf{X} = \mathbf{XAA}'$[3] is the projection of the row vectors in $\mathbf{X}$ onto the plane spanned by $\mathbf{A}$. Thus it assumes that surface $\mathbf{f}$ is linear and it finds a hyperplane that minimizes the sum of squared distances between the observations and their projections onto the

[1] The authors thank Thomas Severini and Richard Briesch for many helpful discussion.

[1] Without loss of generality, we also assume that the columns of $\mathbf{X}$ are mean centered.

[2] See [1] for discussion of engineering applications.

[3] This assumes that $(A'A)^{-1} = I$. For example, A could be orthonormal.

hyperplane. The scores matrix is given by

$$\mathbf{S} = \mathbf{XU}. \tag{3}$$

Several unsupervised learning rules for training neural networks have been shown to converge to the PCA solution.

PCA has been generalized by the the following two methods: principal curves/surfaces (hereafter denoted by PC/S) ([3], [4], and [5]) and NonLinear Principal Components Analysis using autoassociative neural networks (NLPCA) [1]. PC/S was shown to extend PCA by finding a lower-dimensional curve or surface which satisfies a minimum distance property similar to Equation 2. This paper explores the similarities and differences between PC/S and NLPCA and suggests that NLPCA also finds a curve/surface under a slightly different objective function.

2. Nonlinear Feature Extraction

PCA assumes that the relationship between the observed variables and the feature variables is linear, e.g., an increase in the observed variable is associated with a proportional increase in the feature variable. It is easy to imagine situations where this relationship is nonlinear. For example, there could be diminishing returns, where the change in the feature variable is different depending on the value of the observed variable. PC/S and NLPCA extend PCA by relaxing the assumption that $\mathbf{f}$ is linear. This section summarizes the two methods and shows how they are related.

We summarize only the Principal Curves (PC) method. See [3] for information on surfaces. For a given curve $\mathbf{f}$, the *projection index* is

$$s_{\mathbf{f}}(\mathbf{x}) = \sup_{\mathbf{s}}\{\mathbf{s} : \|\mathbf{x} - \mathbf{f}(\mathbf{s})\| = \inf_{\mu} \|\mathbf{x} - \mathbf{f}(\mu)\|\}.$$

The projection index $s_{\mathbf{f}}(\mathbf{x_i})$ evaluated at $\mathbf{x_i}$ is the score value s for which curve $\mathbf{f}(\mathbf{s})$ is closest to $\mathbf{x_i}$. If there are several such values, the largest one is selected. Points which can be projected to more than one closest point on the curve are called *ambiguity points.* Motivated by Equation 2, PC/S estimates $\mathbf{f}$ under the following objective function [4, Proposition 4]:

$$\min_{\mathbf{f}} \sum_{i=1}^{n} \|\mathbf{x_i} - \mathbf{f}(\mathbf{s_f}(\mathbf{x_i}))\|^{\mathbf{2}}. \tag{4}$$

Note that the composition of functions $\mathbf{f}(\mathbf{s_f}(\mathbf{x_i}))$ gives the p-dimensional coordinates of the projection of $\mathbf{x_i}$ onto curve/surface $\mathbf{f}$. The PC method models $\mathbf{f}$ with a scatterplot smoother and fits its model with the principal curve algorithm [4].

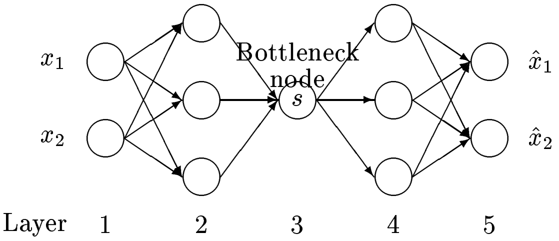

Figure 1: NLPCA neural network architecture.

NLPCA also fits a composition of two functions, $\mathbf{s_f} : \Re^{\mathbf{p}} \to \Re^{\mathbf{r}}$ and $\mathbf{f} : \Re^{\mathbf{r}} \to \Re^{\mathbf{p}}$, which we show are similar, but not identical to the PC/S projection index and curve/surface. The functions are modeled with three-layer neural networks, which can approximate arbitrary smooth functions. A five-layer autoassociative neural network models the composition of functions $\mathbf{f}(\mathbf{s_f}(\mathbf{x_i}))$ and the network is trained under the following objective function:

$$\min_{\mathbf{f}, \mathbf{s_f}} \sum_{i=1}^{n} \|\mathbf{x_i} - \mathbf{f}(\mathbf{s_f}(\mathbf{x_i}))\|^{\mathbf{2}}. \tag{5}$$

Layers 1, 2, and 3 model $\mathbf{s_f}$ and layers 3, 4, and 5 model surface $\mathbf{f}$. 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 nodes in layers 2 and 4 must have nonlinear activation functions to represent arbitrary smooth functions, and the nodes in layers 3 and 5 usually have linear activation functions, although they could be nonlinear. Data compression takes place because the p-dimensional inputs must pass through the $r < p$ dimensional bottleneck layer before reproducing the inputs. Figure 1 shows an example NLPCA network. Once the network has been trained, the bottleneck node activation values give the scores.

PC/S and NLPCA are closely related. Both fit an r-dimensional surface $\mathbf{f}$ and a function $\mathbf{s_f}$ which maps points in $\Re^p$ to scores. When $\mathbf{s_f}$ and $\mathbf{f}$ are linear, both methods produce the PCA solution. The objective functions in Equations 4 and 5 minimize the same function. The difference between the methods are that (1) NLPCA defines $\mathbf{s_f}$ to be continuous whereas the PC/S projection index can be discontinuous, and (2) NLPCA minimizes its objective function over functions $\mathbf{s_f}$ and $\mathbf{f}$ simultaneously while PC/S minimizes over $\mathbf{f}$ and "plugs in" an optimal $\mathbf{s_f}$. In [6] we conjecture that if $\mathbf{s_f}$ were allowed to be discontinuous, the objective function values would be equal. This conjecture has several implications: (1) the NLPCA $\mathbf{s_f}$ would converge to the PC/S $\mathbf{s_f}$; (2) NLPCA (with discontinuous $\mathbf{s_f}$) extracts principal curves/surfaces; and (3) when a principal curve/surface is unique, the PC/S and NLPCA composition of functions $\mathbf{f}(\mathbf{s_f}(\mathbf{x}))$ would be equal for almost every $\mathbf{x}$[4] and therefore the fitted curves/surfaces would differ only by their parameterizations.

3. Continuous Projection Index

An important difference between NLPCA and PC/S is that PC/S allows a discontinuous $\mathbf{s_f}$ while NLPCA defines $\mathbf{s_f}$ to be continuous. We show that this difference can cause suboptimal "projections" which inflate the objective function value and reduce the class of curves and surfaces that can be modeled by NLPCA.

3.1. Suboptimal Projections

The assumption that $\mathbf{s_f}$ is continuous (on $\Re^p$) causes suboptimal "projections" for certain $\mathbf{x}$ values when the curve/surface is nonlinear. A projection is suboptimal when an $\mathbf{x}$ is mapped to a point on the curve other than the point which is closest to it. Consider an example. Suppose we are estimating the parabola

$$\mathbf{x = f(s) = (s, s^2)'} \qquad (6)$$

(i.e., $x_2 = x_1^2$) shown in Figure 2. The ambiguity points of this curve are $\{\mathbf{x : x_1 = 0}$ and $\mathbf{x_2 > 0}\}$ (the axis of symmetry). If we approach an ambiguity point in the direction of a normal through the curve, an optimal $\mathbf{s_f}$ must be discontinuous when we cross the ambiguity point. Figure 2 shows normals drawn through the points $(0.5, 0.25)$ and $(-0.5, 0.25)$ which intersect at ambiguity point $(0, 0.75)$. An optimal $\mathbf{s_f}$ projects each point on the normals to either $(0.5, 0.25)$[5] or $(-0.5, 0.25)$. We fitted an NLPCA model to this parabola problem (with noise added to x_1 and x_2) and the lines in Figure 2 show where the model projects points along the normals. The projections are good for points below the parabola ($\mathbf{x : x_2 < x_1^2}$) and for other points which are fairly close to the curve. As we approach an ambiguity point, the plot shows a suboptimal "fanning behavior" culminating with the ambiguity point projected (suboptimally) around $(0, 0)$. The reason for the fanning is that NLPCA must avoid being discontinuous at the ambiguity point.

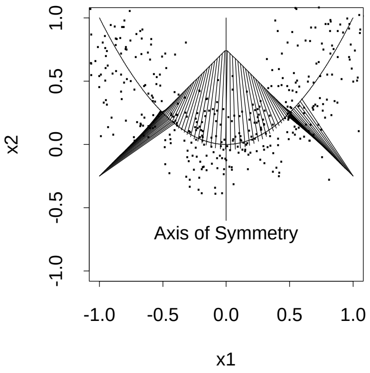

Figure 2: NLPCA fit of parabola with noise.

[3] also examines some related questions and shows that if $\mathbf{s_f}$ is continuous at a point $\mathbf{x}$, then $\mathbf{x}$ is not an ambiguity point (Theorem 4.4). We propose a corallary to this result: if $\mathbf{s_f}$ is assumed to be continuous, it cannot be defined at ambiguity points. Neural networks are defined for each point in $\Re^p$ and therefore must make suboptimal projections to avoid having ambiguity points. The PC method does not have this problem.

3.2. Reduced Class of Curves

Methods which model the projection index with a continuous function cannot approximate curves/surfaces which intersect themselves. The reason for this can be easily understood through an example. A circle in $\Re^2$ intersects itself and can be described by polar coordinates:

$$\mathbf{x = f(s)} = \begin{pmatrix} \cos s \\ \sin s \end{pmatrix}, \qquad (7)$$

where $s \in [0, 2\pi)$. When $\mathbf{s_f}$ is defined to be continuous, a small change in $\mathbf{x}$ values must result in a small change in s values. This is clearly not the case around the point $(1, 0)$, where s jumps from values near 2π to values near 0.

[4] The composition of functions would not necessarily be equal for ambiguity points. [3] shows that the set of ambiguity points has measure zero.

[5] More precisely, $\mathbf{f(s_f(x)) = (0.5, 0.25)'}$ for all $\mathbf{x} \in \{\mathbf{x} :$ $\mathbf{x_1 > 0}$ and $\mathbf{(x_2 - 0.25)/(x_1 - 0.5) = -1}\}$.

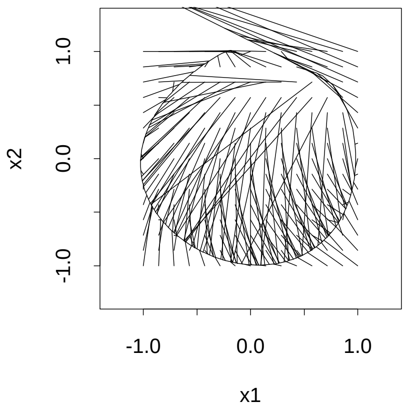

Figure 3: NLPCA fit of full circle.

Asking NLPCA to do something which it cannot do, e.g., model a curve which intersects itself, can produce some strange results, as we illustrate in Figure 3. We generated 100 equally-spaced s values in the interval $[0, 2\pi]$ and evaluated these points in Equation 7 to give 100 points along the perimeter of a circle. We fitted an NLPCA model and then evaluated the model on a cross-validation grid of points in $\Re^2$ to understand the solution. To insure thorough training, we trained the network for 6000 iterations with the quasi-Newton LBFGS nonlinear optimization algorithm[6] [7] and the resulting Fraction of Variance Unexplained (FVU)[7] was 0.0097, although since the data were noiseless, one would have expected FVU ≈ 0. The curve in Figure 3 is the NLPCA approximation and shows that NLPCA does a good job of extracting a circle, except around $\pi/2$, where the ends of the curve repel each other. Most of the training FVU results from the poor fit around $\pi/2$. The projections on the cross validation grid are bad, particularly in the first quadrant, where the $\mathbf{s_f}$ becomes highly nonlinear and sprays its "projections" around the entire perimeter of the circle. The "projections" are particularly bad along the positive part of line $x_1 = 1$, where the projections extend off the plot. An optimal $\mathbf{s_f}$ would project all the points in the first quadrant to points on the circle in the first quadrant, but these projections are clearly sub-optimal. This restriction is not likely to cause problems in practice since a curve will intersect itself only when the relationship among the observed variables is very nonlinear, and it is difficult to imagine when this extreme nonlinearity would occur.

4. Factor Score Identification

In this subsection we discuss how any two parameterizations of a curve ($r = 1$) can be related by a monotone transformation. An implication of this result is that there is an ordinal relationship between any two parameterizations of a curve and hence NLPCA and PC will estimate the order of a feature variable correctly. The PC method parameterizes its curves by arc length, so the magnitude of differences between score values can also be interpreted. [6] shows how to modify NLPCA to produce an arc-length parameterization.

Suppose that the projection index and curve of a one-dimensional NLPCA model are given by, $\mathbf{s} = \mathbf{s_f}(\mathbf{x})$ and $\hat{\mathbf{x}} = \mathbf{f}(\mathbf{s})$, respectively. Let ψ be an invertible function with inverse ψ^{-1}. For example, ψ could be the logarithm function and ψ^{-1} could be the exponential function. The value of the objective function (Equation 5) is unchanged by applying ψ to the factor scores because

$$\mathbf{f}(\mathbf{s_f}(\mathbf{x_i})) = \mathbf{f}(\psi^{-\mathbf{1}}(\psi(\mathbf{s_f}(\mathbf{x_i})))) \qquad (8)$$

for each i. The factor scores for the transformed model are $\mathbf{s}^* = \psi(\mathbf{s_f}(\mathbf{X})) = \mathbf{s}^*_{\mathbf{f}}(\mathbf{X})$. Thus it is possible to produce the same objective function value with different factor scores. In [6] we show that the converse of this statement is also true, i.e., if $\mathbf{f}(\mathbf{s_f}(\mathbf{x})) = \mathbf{f}^*(\mathbf{s_{f^*}}(\mathbf{x}))$, then there exists invertible function ψ so that $\psi(\mathbf{s_f}(\mathbf{x})) = \mathbf{f}^*(\mathbf{x})$[8]. Since invertible functions are monotone, the *order* of the score values is estimated by nonlinear feature extraction methods.

5. Examples

This section describes a small simulation study to compare NLPCA and PC. The study presented here is small in that we examine only two problems, but the results are representative of our experiences in general with PCs and NLPCA on

[6] We thank Jorge Nocedal for making his code available.

[7] $\mathrm{FVU} = \sum\sum(x_{ij} - \hat{x}_{ij})^2 / \sum\sum(x_{ij} - \bar{x}_{.j})^2$, where $\bar{x}_{.j}$ is the sample mean of column j.

[8] Curve $\mathbf{f}$ must also be 1-1, so that it does not intersect itself.

roughly 20 different problems. We fitted PC and NLPCA models to a parabola (Eq. 6) and a 3/4 circle (Eq. 7 with $s \in [0, 3\pi/2]$) after adding 3 different amounts of noise. We chose a 3/4 circle because it does not intersect itself and it cannot be parameterized by the lengths of projections onto some line. The parabola and a semicircle ($s \in [0, \pi]$) can be parameterized by projections onto the x_1 axis and thus PCA also should estimate the order of the feature correctly in noiseless situations. We generated $n = 100$ s values from $\mathcal{U}[-1, 1]$ for the parabola and $\mathcal{U}[0, 3\pi/2]$ for the 3/4 circle. We then added Gaussian noise to the $\mathbf{x}$ vectors with standard deviations 0, 0.1, and 0.2. We fitted NLPCA models with a neural network simulator developed by us and PC using Hastie's S function[9] with the splines option. The fits were evaluated on a cross-validation data set of 1000 equally-spaced s values over the respective intervals and the FVU values are given in Table 1. Note that this cross-validation procedure measures how well the curve has been approximated, but does not measure the quality of the projection index.

	3/4 Circle		**Parabola**	
σ_ϵ	**NLPCA**	**PC**	**NLPCA**	**PC**
0	0.0017	0.0037	0.0000	0.0008
0.1	0.0009	0.0050	0.0033	0.0025
0.2	0.0324	0.0028	0.0138	0.0042

Table 1: FVU Values comparing NLPCA and PCs.

NLPCA seems to do marginally better in the noiseless problem, but both models give good fits. The reason why PC does worse is that they give a conservative estimate for the ends of a curve (see [4, Fig. 4]). NLPCA gives a better approximation to the ends in these problems where the ends are smooth continuations of the rest of the curve. PC seems to be less affected by noise, whereas the NLPCA fits with $\sigma_\epsilon = 0.2$ are several times worse than those with $\sigma_\epsilon = 0.1$. Both methods estimate the order of score values correctly.

6. Conclusions

NLPCA and PC/S estimate their respective $\mathbf{f}$ and $\mathbf{s_f}$ under similar objective functions. NLPCA defines $\mathbf{s_f}$ to be continuous; without this assumption, we conjecture NLPCA would produce principal curves/surfaces. Our empirical experience suggests that both NLPCA and PC give good approximations to underlying nonlinear features. The performance of NLPCA seems to degrade with noisy data while PC appears more robust to noise. The PC estimate is conservative at the ends of a curve (since there are less data). NLPCA seems to give a good approximation to the ends in nonpathological problems. NLPCA can give suboptimal projections for points requiring cross validation (particularly those near ambiguity points) but PC does not have this problem. NLPCA cannot approximate curves which intersect themselves. Any two parameterizations of a curve can be related by a monotone transformation and therefore NLPCA estimates the order of a feature variable correctly. PC provides an arc-length parameterization which can be more interpretable since the magnitude of a difference between score values can be interpreted. NLPCA models can be fitted using most neural network simulators (commercial and public domain). An S implementation of PC is available from an anonymous ftp site. See [6] for further discussion of NLPCA, including its performance on surfaces and sequential NLPCA.

References

[1] M.A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[2] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis.* Academic Press, London, 1979.

[3] T. Hastie. *Principal Curves and Surfaces.* PhD thesis, Stanford University, November 1984.

[4] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Society*, 84(406):502–516, 1989.

[5] M. LeBlanc and R. Tibshirani. Adaptive principal surfaces. *Journal of the American Statistical Society*, 89(425):53–64, 1994.

[6] E.C. Malthouse. *Nonlinear Partial Least Squares.* PhD thesis, Northwestern University, June 1995.

[7] D Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[9] We thank Hastie for making this available.