1

Developments and Applications of Nonlinear Principal Component Analysis – a Review

Uwe Kruger[1], Junping Zhang[2], and Lei Xie[3]

[1] School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT9 5AH, UK,
uwe.kruger@ee.qub.ac.uk
[2] Department of Computer Science and Engineering, Fudan University, Shanghai 200433, P.R. China,
jpzhang@fudan.edu.cn
[3] National Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, P.R. China,
leix@iipc.zju.edu.cn

Summary. Although linear principal component analysis (PCA) originates from the work of Sylvester [67] and Pearson [51], the development of nonlinear counterparts has only received attention from the 1980s. Work on nonlinear PCA, or NLPCA, can be divided into the utilization of autoassociative neural networks, principal curves and manifolds, kernel approaches or the combination of these approaches. This article reviews existing algorithmic work, shows how a given data set can be examined to determine whether a conceptually more demanding NLPCA model is required and lists developments of NLPCA algorithms. Finally, the paper outlines problem areas and challenges that require future work to mature the NLPCA research field.

1.1 Introduction

PCA is a data analysis technique that relies on a simple transformation of recorded observation, stored in a vector $\mathbf{z} \in \mathbb{R}^N$, to produce statistically independent score variables, stored in $\mathbf{t} \in \mathbb{R}^n$, $n \leq N$:

$$\mathbf{t} = \mathbf{P}^T \mathbf{z} \; . \tag{1.1}$$

Here, $\mathbf{P}$ is a transformation matrix, constructed from *orthonormal* column vectors. Since the first applications of PCA [21], this technique has found its way into a wide range of different application areas, for example signal processing [75], factor analysis [29, 44], system identification [77], chemometrics [20, 66] and more recently, general data mining [11, 70, 58] including image processing [17, 72] and pattern recognition [47, 10], as well as process

monitoring and quality control [1, 82] including multiway [48], multiblock [52] and multiscale [3] extensions. This success is mainly related to the ability of PCA to describe significant information/variation within the recorded data typically by the first few score variables, which simplifies data analysis tasks accordingly.

Sylvester [67] formulated the idea behind PCA, in his work the removal of redundancy in bilinear quantics, that are polynomial expressions where the sum of the exponents are of an order greater than 2, and Pearson [51] laid the conceptual basis for PCA by defining lines and planes in a multivariable space that present the closest fit to a given set of points. Hotelling [28] then refined this formulation to that used today. Numerically, PCA is closely related to an eigenvector-eigenvalue decomposition of a data covariance, or correlation matrix and numerical algorithms to obtain this decomposition include the iterative NIPALS algorithm [78], which was defined similarly by Fisher and MacKenzie earlier on [80], and the singular value decomposition. Good overviews concerning PCA are given in Mardia *et al.* [45], Joliffe [32], Wold *et al.* [80] and Jackson [30].

The aim of this article is to review and examine nonlinear extensions of PCA that have been proposed over the past two decades. This is an important research field, as the application of linear PCA to nonlinear data may be inadequate [49]. The first attempts to present nonlinear PCA extensions include a generalization, utilizing a nonmetric scaling, that produces a nonlinear optimization problem [42] and constructing a curves through a given cloud of points, referred to as principal curves [25]. Inspired by the fact that the reconstruction of the original variables, $\widehat{\mathbf{z}}$ is given by:

$$\widehat{\mathbf{z}} = \mathbf{P}\mathbf{t} = \overbrace{\mathbf{P}\underbrace{\left(\mathbf{P}^T\mathbf{z}\right)}_{\text{mapping}}}^{\text{demapping}}, \tag{1.2}$$

that includes the determination of the score variables (mapping stage) and the determination of $\widehat{\mathbf{z}}$ (demapping stage), Kramer [37] proposed an *autoassociative neural network* (ANN) structure that defines the mapping and demapping stages by neural network layers. Tan and Mavrovouniotis [68] pointed out, however, that the 5 layers network topology of autoassociative neural networks may be difficult to train, i.e. network weights are difficult to determine if the number of layers increases [27].

To reduce the network complexity, Tan and Mavrovouniotis proposed an *input training* (IT) network topology, which omits the mapping layer. Thus, only a 3 layer network remains, where the reduced set of nonlinear principal components are obtained as part of the training procedure for establishing the IT network. Dong and McAvoy [16] introduced an alternative approach that divides the 5 layer autoassociative network topology into two 3 layer topologies, which, in turn, represent the nonlinear mapping and demapping functions. The output of the first network, that is the mapping layer, are

the score variables which are determined using the principal curve approach. The second layer then represents the demapping function for which the score variables are the inputs and the original variables are the outputs. Jia *et al.* [31] presented a critical review of the techniques in references [68, 16] and argued that the incorporation of a principal curve algorithm into a neural network structure [16] may only cover a limited class of nonlinear functions. Hence, the IT network topology [68] may provide a more effective nonlinear compression than the technique by Dong and McAvoy [16]. In addition, Jia *et al.* [31] further refined the IT concept by introducing a linear compression using PCA first, which is followed by the application of the IT algorithm using the scaled linear principal components.

More recently, *Kernel PCA* (KPCA) has been proposed by Schölkopf [57, 56]. KPCA first maps the original variable set $\mathbf{z}$ onto a high-dimensional feature space using the mapping function $\mathbf{\Phi}(\mathbf{z})$. Then, KPCA performs a conventional linear principal component analysis on $\mathbf{\Phi}(\mathbf{z})$. The KPCA approach takes advantage of the fact that the mapping function $\mathbf{z} \mapsto \mathbf{\Phi}(\mathbf{z})$ does not need to be known *a priori*. Furthermore, this mapping function can be approximated using Kernel functions in a similar fashion to a radial basis function neural network. In fact, the identification of a KPCA model utilizes scalar products of the observations, which are then nonlinearly transformed using Kernel functions. This presents a considerable advantage over neural network approaches since no nonlinear optimization procedure needs to be considered. Resulting from this conceptual simplicity and computational efficiency, KPCA has recently found its way into a wide range of applications, most notably in the areas of face recognition [36], image de-noising [40] and industrial process fault detection [12, 81].

This article is divided into the following sections. A brief review of PCA including its most important properties is given next, prior to the introduction of a nonlinearity test. Section 4 then details nonlinear extensions of PCA. Section 5 then critically evaluates existing work on NLPCA in terms of computational demand in computing a model as well as generalization issues and provides a roadmap for future research work.

1.2 PCA Preliminaries

PCA analyses a data matrix $\mathbf{Z} \in \mathbb{R}^{K \times N}$ that possesses the following structure:

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{21} & z_{13} & \cdots & z_{1j} & \cdots & z_{1N} \\ z_{21} & z_{22} & z_{23} & \cdots & z_{2j} & \cdots & z_{2N} \\ z_{31} & z_{32} & z_{33} & \cdots & z_{3j} & \cdots & z_{3N} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ z_{i1} & z_{i2} & z_{i3} & \cdots & z_{ij} & \cdots & z_{iN} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ z_{K-1,1} & z_{K-1,2} & z_{K-1,3} & \cdots & z_{K-1,j} & \cdots & z_{K-1,N} \\ z_{K1} & z_{K2} & z_{K3} & \cdots & z_{Kj} & \cdots & z_{KN} \end{bmatrix}, \tag{1.3}$$

where N and K are the number of recorded variables and the number of available observations, respectively. Defining the rows and columns of $\mathbf{Z}$ by vectors $\mathbf{z}_i \in \mathbb{R}^N$ and $\boldsymbol{\zeta}_j \in \mathbb{R}^K$, respectively, $\mathbf{Z}$ can be rewritten as shown below:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \mathbf{z}_3^T \\ \vdots \\ \mathbf{z}_i^T \\ \vdots \\ \mathbf{z}_{K-1}^T \\ \mathbf{z}_K^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{\zeta}_1 \; \boldsymbol{\zeta}_2 \; \boldsymbol{\zeta}_3 \cdots \boldsymbol{\zeta}_j \cdots \boldsymbol{\zeta}_N \end{bmatrix} . \tag{1.4}$$

The first and second order statistics of the original set variables $\mathbf{z}^T = \begin{pmatrix} z_1 \; z_2 \; z_3 \cdots \;\; z_j \cdots z_N \end{pmatrix}$ are:

$$E\{\mathbf{z}\} = \mathbf{0} \qquad E\left\{\mathbf{z}\mathbf{z}^T\right\} = \mathbf{S}_{ZZ} \tag{1.5}$$

with the correlation matrix of $\mathbf{z}$ being defined as $\mathbf{R}_{ZZ}$.

The PCA analysis entails the determination of a set of score variables t_k, $k \in \left\{ 1 \; 2 \; 3 \cdots n \right\}$, $n \leq N$, by applying a linear transformation of $\mathbf{z}$:

$$t_k = \sum_{j=1}^{N} p_{kj} z_j \tag{1.6}$$

under the following constraint for the parameter vector

$$\mathbf{p}_k^T = \begin{pmatrix} p_{k1} \; p_{k2} \; p_{k3} \cdots p_{kj} \cdots p_{kN} \end{pmatrix} :$$

$$\sqrt{\sum_{j=1}^{N} p_{kj}^2} = \|\mathbf{p}_k\|_2 = 1 \; . \tag{1.7}$$

Storing the score variables in a vector $\mathbf{t}^T = \begin{pmatrix} t_1 \; t_2 \; t_3 \cdots t_j \cdots t_n \end{pmatrix}$, $\mathbf{t} \in \mathbb{R}^n$ has the following first and second order statistics:

$$E\left\{\mathbf{t}\right\} = \mathbf{0} \qquad E\left\{\mathbf{t}\mathbf{t}^T\right\} = \mathbf{\Lambda}\ , \tag{1.8}$$

where $\mathbf{\Lambda}$ is a diagonal matrix. An important property of PCA is that the variance of the score variables represent the following maximum:

$$\lambda_k = \arg\max_{\mathbf{p}_k}\left\{E\left\{t_k^2\right\}\right\} = \arg\max_{\mathbf{p}_k}\left\{E\left\{\mathbf{p}_k^T\mathbf{z}\mathbf{z}^T\mathbf{p}_k\right\}\right\}\ , \tag{1.9}$$

that is constraint by:

$$E\left\{\begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ \vdots \\ t_{k-1} \end{pmatrix} t_k\right\} = \mathbf{0} \qquad \|\mathbf{p}_k\|_2^2 - 1 = 0\ . \tag{1.10}$$

Anderson [2] indicated that the formulation of the above constrained optimization can alternatively be written as:

$$\lambda_k = \arg\max_{\mathbf{p}}\left\{E\left\{\mathbf{p}^T\mathbf{z}\mathbf{z}^T\mathbf{p}\right\} - \lambda_k\left(\mathbf{p}^T\mathbf{p} - 1\right)\right\} \tag{1.11}$$

under the assumption that λ_k is predetermined. Reformulating (1.11) to determine $\mathbf{p}_k$ gives rise to:

$$\mathbf{p}_k = \arg\frac{\partial}{\partial\mathbf{p}}\left\{E\left\{\mathbf{p}^T\mathbf{z}\mathbf{z}^T\mathbf{p}\right\} - \lambda_k\left(\mathbf{p}^T\mathbf{p} - 1\right)\right\} = \mathbf{0} \tag{1.12}$$

and produces

$$\mathbf{p}_k = \arg\left\{E\left\{\mathbf{z}\mathbf{z}^T\right\}\mathbf{p} - 2\lambda_k\mathbf{p}\right\} = \mathbf{0}\ . \tag{1.13}$$

Incorporating (1.5) allows constructing an analytical solution of this constrained optimization problem:

$$\left[\mathbf{S}_{ZZ} - \lambda_k\mathbf{I}\right]\mathbf{p}_k = \mathbf{0}\ , \tag{1.14}$$

which implies that the kth largest eigenvalue of $\mathbf{S}_{ZZ}$ is the variance of the kth score variable and the parameter vector $\mathbf{p}_k$, associated with λ_k, stores the kth set of coefficients to obtain the kth linear transformation of the original variable set $\mathbf{z}$ to produce t_k. Furthermore, given that $\mathbf{S}_{ZZ}$ is a positive definite or semidefinite matrix it follows that the eigenvalues are positive and real and the eigenvectors are mutually orthonormal. The solution of Equation (1.14) also implies that the score variables are statistically independent, as defined in (1.10), which follows from:

$$\widehat{\mathbf{S}}_{ZZ} = \frac{1}{K-1}\widehat{\mathbf{Z}}^T\widehat{\mathbf{Z}} = \widehat{\mathbf{P}}\widehat{\mathbf{\Lambda}}\widehat{\mathbf{P}}^T \Longrightarrow \frac{1}{K-1}\widehat{\mathbf{P}}^T\mathbf{Z}^T\mathbf{Z}\widehat{\mathbf{P}} = \frac{1}{K-1}\widehat{\mathbf{T}}^T\widehat{\mathbf{T}} = \widehat{\mathbf{\Lambda}}\ . \tag{1.15}$$

Here, the index $\widehat{\circ}$ represents estimates of the covariance matrix, its eigenvectors and eigenvalues and the score matrix using the reference data stored in $\mathbf{Z}$. A solution of Equations (1.9) and (1.10) can be obtained using a singular value decomposition of the data covariance matrix $\widehat{\mathbf{S}}_{ZZ}$ or the iterative Power method [22].

1.3 Nonlinearity Test for PCA Models

This section discusses how to determine whether the underlying structure within the recorded data is linear or nonlinear. Kruger *et al.* [38] introduced this nonlinearity test using the principle outlined in Figure 1.1. The left plot in

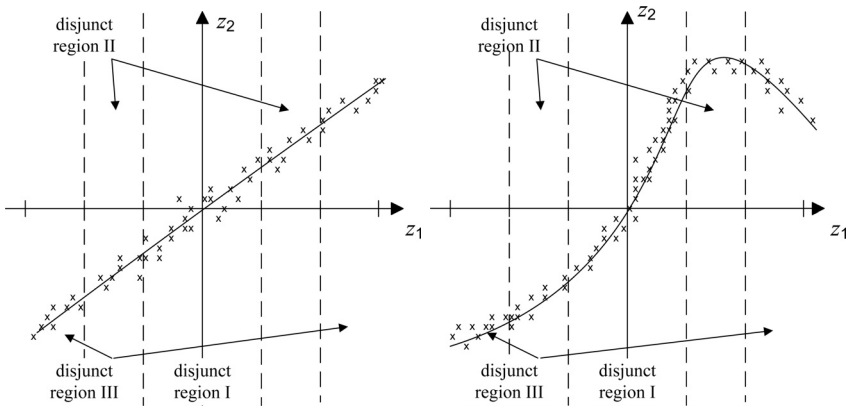


Fig. 1.1. Principle of nonlinearity test

this figure shows that the first principal component describes the underlying linear relationship between the two variables, z_1 and z_2, while the right plot describes some basic nonlinear function, indicated by the curve.

By dividing the operating region into several disjunct regions, where the first region is centered around the origin of the coordinate system, a PCA model can be obtained from the data of each of these disjunct regions. With respect to Figure 1.1, this would produce a total of 3 PCA models for each disjunct region in both cases, the linear (left plot) and the nonlinear case (right plot). To determine whether a linear or nonlinear variable interrelationship can be extracted from the data, the principle idea is to take advantage of the residual variance in each of the regions. More precisely, *accuracy bounds* that are based on the residual variance are obtained for *one of the PCA models*, for example that of disjunct region I, and the residual variance of the *remaining PCA models* (for disjunct regions II and III) are benchmarked against *these bounds*. The test is completed if *each of the PCA models* has been used to determine accuracy bounds which are then benchmarked against the residual variance of the respective *remaining PCA models*.

The reason of using the residual variance instead of the variance of the retained score variables is as follows. The residual variance is independent of the region if the underlying interrelationship between the original variables is linear, which the left plot in Figure 1.1 indicates. In contrast, observations that have a larger distance from the origin of the coordinate system will, by default, produce a larger projection distance from the origin, that is a

larger score value. In this respect, observations that are associated with an adjunct region that are further outside will logically produce a larger variance irrespective of whether the variable interrelationships are linear or nonlinear.

The detailed presentation of the nonlinearity test in the remainder of this section is structured as follows. Next, the assumptions imposed on the nonlinearity test are shown, prior to a detailed discussion into the construction of disjunct regions. Subsection 3.3 then shows how to obtain statistical confidence limits for the nondiagonal elements of the correlation matrix. This is followed by the definition of the accuracy bounds. Finally, a summary of the nonlinearity test is presented and some example studies are presented to demonstrate the working of this test.

1.3.1 Assumptions

The assumptions imposed on the nonlinearity test are summarized below [38].

1. The variables are mean-centered and scaled to unit variance with respect to disjunct regions for which the accuracy bounds are to be determined.
2. Each disjunct region has the same number of observations.
3. A PCA model is determined for one region where the the accuracy bounds describe the variation for the sum of the discarded eigenvalues in that region.
4. PCA models are determined for the remaining disjunct regions.
5. The PCA models for each region include the same number of retained principal components.

1.3.2 Disjunct Regions

Here, we investigate how to construct the disjunct regions and how many disjunct regions should be considered. In essence, dividing the operating range into the disjunct regions can be carried out through prior knowledge of the process or by directly analyzing the recorded data. Utilizing *a priori* knowledge into the construction of the disjunct regions, for example, entails the incorporation of knowledge about distinct operating regions of the process. A direct analysis, on the other hand, by applying scatter plots of the first few retained principal components could reveal patterns that are indicative of distinct operating conditions. Wold *et al.* [80], page 46, presented an example of this based on a set of 20 "natural" amino acids.

If the above analysis does not yield any distinctive features, however, the original operating region could be divided into two disjunct regions initially. The nonlinearity test can then be applied to these two initial disjunct regions. Then, the number of regions can be increased incrementally, followed by a subsequent application of the test. It should be noted, however, that increasing the number of disjunct regions is accompanied by a reduction in the number of observations in each region. As outlined the next subsection, a sufficient

number of observations are required in order to prevent large Type I and II errors for testing the hypothesis of *using a linear model* against the alternative hypothesis of *rejecting that a linear model can be used*.

Next, we discuss which of the disjunct regions should be used to establish the accuracy bounds. Intuitively, one could consider the most centered region for this purpose or alternatively, a region that is at the margin of the original operating region. More practically, the region at which the process is known to operate most often could be selected. This, however, would require *a priori* knowledge of the process. However, a simpler approach relies on the incorporation of the cross-validation principle [65, 64] to automate this selection. In relation to PCA, cross-validation has been proposed as a technique to determine the number of retained principal components by Wold [79] and Krzanowski [39].

Applied to the nonlinearity test, the cross-validation principle could be applied in the following manner. First, select one disjunct region and compute the accuracy bounds of that region. Then, benchmark the residual variance of the remaining PCA models against this set of bounds. The test is completed if accuracy bounds have been computed for each of the disjunct regions and the residual variances of the PCA models of the respective remaining disjunct regions have been benchmarked against these accuracy bounds. For example, if 3 disjunct regions are established, the PCA model of the first region is used to calculate accuracy bounds and the residual variances of the 3 PCA models (one for each region) is benchmarked against this set of bounds. Then, the PCA model for the second region is used to determine accuracy bounds and again, the residual variances of the 3 PCA models are benchmarked against the second set of bounds. Finally, accuracy bounds for the PCA model of the 3rd region are constructed and each residual variance is compared to this 3rd set of bounds. It is important to note that the PCA models will vary depending upon which region is currently used to compute accuracy bounds. This is a result of the normalization procedure, since the mean and variance of each variable may change from region to region.

1.3.3 Confidence Limits for Correlation Matrix

The data correlation matrix, which is symmetric and positive semidefinite, for a given set of N variables has the following structure:

$$\mathbf{R}_{ZZ} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1N} \\ r_{21} & 1 & \cdots & r_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & \cdots & 1 \end{bmatrix} . \tag{1.16}$$

Given that the total number of disjunct regions is m the number of observations used to construct any correlation matrix is $\widetilde{K} = K/m$, rounded to the nearest integer. Furthermore, the correlation matrix for constructing the

PCA model for the hth disjunct region, which is utilized to determine of the accuracy bound, is further defined by $\mathbf{R}_{ZZ}^{(h)}$. Whilst the diagonal elements of this matrix are equal to one, the nondiagonal elements represent correlation coefficients for which confidence limits can be determined as follows:

$$r_{ij}^{(h)} = \frac{\exp\left(2\varsigma_{ij}^{(h)}\right) - 1}{\exp\left(2\varsigma_{ij}^{(h)}\right) + 1} \text{ if } i \neq j \ , \tag{1.17}$$

where $\varsigma_{ij}^{(h)} = \varsigma_{ij}^{(h)^*} \pm \varepsilon$, $\varsigma_{ij}^{(h)^*} = \ln\left(1 + r_{ij}^{(h)^*} \Big/ 1 - r_{ij}^{(h)^*}\right) \Big/ 2$, $r_{ij}^{(h)^*}$ is the sample correlation coefficient between the ith and jth process variable, $\varepsilon = {}^{c_\alpha} \Big/ \sqrt{\widetilde{K} - 3}$ and c_α is the critical value of a normal distribution with zero mean, unit variance and a significance level α. This produces two confidence limits for each of the nondiagonal elements of $\mathbf{R}_{ZZ}^{(h)}$, which implies that the estimate nondiagonal elements with a significance level of α, is between

$$\mathbf{R}_{ZZ}^{(h)} = \left[\begin{array}{c|c|c|c} 1 & r_{12_L}^{(h)} \leq r_{12}^{(h)} \leq r_{12_U}^{(h)} & \cdots & r_{1N_L}^{(h)} \leq r_{1N}^{(h)} \leq r_{1N_U}^{(h)} \\ \hline r_{21_L}^{(h)} \leq r_{21}^{(h)} \leq r_{21_U}^{(h)} & 1 & \cdots & r_{2N_L}^{(h)} \leq r_{2N}^{(h)} \leq r_{2N_U}^{(h)} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline r_{N1_L}^{(h)} \leq r_{N1}^{(h)} \leq r_{N1_U}^{(h)} & r_{N2_L}^{(h)} \leq r_{N2}^{(h)} \leq r_{N2_U}^{(h)} & \cdots & 1 \end{array}\right] , \tag{1.18}$$

where the indices U and L refer to the upper and lower confidence limit, that is $r_{ij_L}^{(h)} = \frac{\exp\left(2\left(\varsigma_{ij}^{(h)} - \varepsilon\right)\right) - 1}{\exp\left(2\left(\varsigma_{ij}^{(h)} - \varepsilon\right)\right) + 1}$ and $r_{ij_U}^{(h)} = \frac{\exp\left(2\left(\varsigma_{ij}^{(h)} + \varepsilon\right)\right) - 1}{\exp\left(2\left(\varsigma_{ij}^{(h)} + \varepsilon\right)\right) + 1}$. A simplified version of Equation (1.18) is shown below

$$\mathbf{R}_{ZZ_L}^{(h)} \leq \mathbf{R}_{ZZ}^{(h)} \leq \mathbf{R}_{ZZ_U}^{(h)} \tag{1.19}$$

which is valid elementwise. Here, $\mathbf{R}_{ZZ_L}^{(h)}$ and $\mathbf{R}_{ZZ_U}^{(h)}$ are matrices storing the lower confidence limits and the upper confidence limits of the nondiagonal elements, respectively.

It should be noted that the confidence limits for each correlation coefficient is dependent upon the number of observations contained in each disjunct region, $\widetilde{K}$. More precisely, if $\widetilde{K}$ reduces the confidence region widens according to (1.17). This, in turn, undermines the sensitivity of this test. It is therefore important to record a sufficiently large reference set from the analyzed process in order to (i) guarantee that the number of observations in each disjunct region does not produce excessively wide confidence regions for each correlation coefficient, (ii) produce enough disjunct regions for the test and (iii) extract information encapsulated in the recorded observations.

1.3.4 Accuracy Bounds

Finally, (1.19) can now be taken advantage of in constructing the accuracy bounds for the hth disjunct region. The variance of the residuals can be calculated based on the Frobenius norm of the residual matrix $\mathbf{E}_h$. Beginning with the PCA decomposition of the data matrix $\mathbf{Z}_h$, storing the observations of the hth disjunct region, into the product of the associated score and loading matrices, $\mathbf{T}_h\mathbf{P}_h^T$ and the residual matrix $\mathbf{E}_h = \mathbf{T}_h^*\mathbf{P}_h^{*^T}$:

$$\mathbf{Z}_h = \mathbf{T}_h\mathbf{P}_h^T + \mathbf{E}_h = \mathbf{T}_h\mathbf{P}_h^T + \mathbf{T}_h^*\mathbf{P}_h^{*^T} \ , \tag{1.20}$$

the sum of the residual variances for each original variable, ρ_{i_h}, $\rho_h = \sum_{i=1}^{N} \rho_{i_h}$ can be determined as follows:

$$\rho_h = \frac{1}{\widetilde{K}-1}\sum_{i=1}^{\widetilde{K}}\sum_{j=1}^{N} e_{ij_h}^2 = \frac{1}{\widetilde{K}-1}\|\mathbf{E}_h\|_2^2. \tag{1.21}$$

which can be simplified to:

$$\rho_h = \frac{1}{\widetilde{K}-1}\|\mathbf{T}_h^*\mathbf{P}_h^{*^T}\|_2^2 = \frac{1}{\widetilde{K}-1}\|\mathbf{U}_h^*\mathbf{\Lambda}_h^{*^{1/2}}\sqrt{\widetilde{K}-1}\mathbf{P}_h^{*^T}\|_2^2 \tag{1.22}$$

and is equal to:

$$\rho_h = \frac{\widetilde{K}-1}{\widetilde{K}-1}\|\mathbf{\Lambda}_h^{*^{1/2}}\|_2^2 = \sum_{i=n+1}^{N}\lambda_i \ . \tag{1.23}$$

Equations (1.20) and (1.22) utilize a singular value decomposition of $\mathbf{Z}_h$ and reconstructs the discarded components, that is

$$\mathbf{E}_h = \mathbf{U}_h^*\left[\mathbf{\Lambda}_h^*\sqrt{\widetilde{K}-1}\right]\mathbf{P}_h^{*^T} = \mathbf{T}_h^*\mathbf{P}_h^{*^T}.$$

Since $\mathbf{R}_{ZZ}^{(h)} = \begin{bmatrix}\mathbf{P}_h & \mathbf{P}_h^*\end{bmatrix}\begin{bmatrix}\mathbf{\Lambda}_h & \mathbf{0}\\ \mathbf{0} & \mathbf{\Lambda}_h^*\end{bmatrix}\begin{bmatrix}\mathbf{P}_h^T\\ \mathbf{P}_h^{*^T}\end{bmatrix}$, the discarded eigenvalues λ_1, λ_2, …, λ_N depend on the elements in the correlation matrix $\mathbf{R}_{ZZ}$. According to (1.18) and (1.19), however, these values are calculated within a confidence limits obtained for a significance level α. This, in turn, gives rise to the following optimization problem:

$$\begin{aligned}\rho_{h_{\max}} &= \arg\max_{\Delta\mathbf{R}_{ZZ_{\max}}} \rho_h\left(\mathbf{R}_{ZZ} + \Delta\mathbf{R}_{ZZ_{\max}}\right),\\ \rho_{h_{\min}} &= \arg\min_{\Delta\mathbf{R}_{ZZ_{\min}}} \rho_h\left(\mathbf{R}_{ZZ} + \Delta\mathbf{R}_{ZZ_{\min}}\right) \ ,\end{aligned} \tag{1.24}$$

which is subject to the following constraints:

$$\begin{aligned}\mathbf{R}_{ZZ_L} \le \mathbf{R}_{ZZ} + \Delta\mathbf{R}_{ZZ_{\max}} \le \mathbf{R}_{ZZ_U},\\ \mathbf{R}_{ZZ_L} \le \mathbf{R}_{ZZ} + \Delta\mathbf{R}_{ZZ_{\min}} \le \mathbf{R}_{ZZ_U} \ ,\end{aligned} \tag{1.25}$$

where $\Delta\mathbf{R}_{ZZ_{\max}}$ and $\Delta\mathbf{R}_{ZZ_{\min}}$ are perturbations of the nondiagonal elements in $\mathbf{R}_{ZZ}$ that result in the determination of a maximum value, $\rho_{h_{\max}}$, and a minimum value, $\rho_{h_{\min}}$, of ρ_h, respectively.

The maximum and minimum value, $\rho_{h_{\max}}$ and $\rho_{h_{\min}}$, are defined as the accuracy bounds for the hth disjunct region. The interpretation of the accuracy bounds is as follows.

Definition 1. *Any set of observations taken from the same disjunct operating region cannot produce a larger or a smaller residual variance, determined with a significance of α, if the interrelationship between the original variables is linear.*

The solution of Equations (1.24) and (1.25) can be computed using a genetic algorithm [63] or the more recently proposed particle swarm optimization [50].

1.3.5 Summary of the Nonlinearity Test

After determining the accuracy bounds for the hth disjunct region, detailed in the previous subsection, a PCA model is obtained for each of the remaining $m-1$ regions. The sum of the $N-n$ discarded eigenvalues is then benchmarked against these limits to examine whether they fall inside or at least one residual variance value is outside. The test is completed if accuracy bounds have been computed for each of the disjunct regions including a benchmarking of the respective remaining $m-1$ residual variance. If for each of these combinations the residual variance is within the accuracy bound the process *is said to be linear*. In contrast, if at least one of the residual variances is outside one of the accuracy bounds, *it must be concluded that the variable interrelationships are nonlinear*. In the latter case, the uncertainty in the PCA model accuracy is smaller than the variation of the residual variances, implying that a nonlinear PCA model must be employed.

The application of the nonlinearity test involves the following steps.

1. Obtain a sufficiently large set of process data;
2. Determine whether this set can be divided into disjunct regions based on *a priori* knowledge; if yes, goto step 5 else goto step 3;
3. Carry out a PCA analysis of the recorded data, construct scatter diagrams for the first few principal components to determine whether distinctive operating regions can be identified; if so goto step 5 else goto step 4;
4. Divide the data into two disjunct regions, carry out steps 6 to 11 by setting $h = 1$, and investigate whether nonlinearity within the data can be proven; if not, increase the number of disjunct regions incrementally either until the sum of discarded eigenvalues violate the accuracy bounds or the number of observations in each region is insufficient to continue the analysis;
5. Set $h = 1$;
6. Calculate the confidence limits for the nondiagonal elements of the correlation matrix for the hth disjunct region (Equations (1.17) and (1.18));

7. Solve Equations (1.24) and (1.25) to compute accuracy bounds $\sigma_{h_{\max}}$ and $\sigma_{h_{\min}}$;
8. Obtain correlation/covariance matrices for each disjunct region (scaled with respect to the variance of the observations within the hth disjunct region;
9. Carry out a singular value decomposition to determine the sum of eigenvalues for each matrix;
10. Benchmark the sums of eigenvalues against the hth set of accuracy bounds to test the hypothesis that *the interrelationships between the recorded process variables are linear* against the alternative hypothesis that *the variable interrelationships are nonlinear*;
11. if $h = N$ terminate the nonlinearity test else goto step 6 by setting $h = h + 1$.

Examples of how to employ the nonlinearity test is given in the next subsection.

1.3.6 Example Studies

These examples have two variables, z_1 and z_2. They describe (a) a linear interrelationship and (b) a nonlinear interrelationship between z_1 and z_2. The examples involve the simulation of 1000 observations of a single score variable t that stem from a uniform distribution such that the division of this set into 4 disjunct regions produces 250 observations per region. The mean value of t is equal to zero and the observations of t spread between $+4$ and -4.

In the linear example, z_1 and z_2 are defined by superimposing two independently and identically distributed sequences, e_1 and e_2, that follow a normal distribution of zero mean and a variance of 0.005 onto t:

$$z_1 = t + e_1, e_1 = \mathcal{N}\{0, 0.005\} \qquad z_2 = t + e_2, e_2 = \mathcal{N}\{0, 0.005\} \ . \tag{1.26}$$

For the nonlinear example, z_1 and z_2, are defined as follows:

$$z_1 = t + e_1 \qquad z_2 = t^3 + e_2 \tag{1.27}$$

with e_1 and e_2 described above. Figure 1.2 shows the resultant scatter plots for the linear example (right plot) and the nonlinear example (left plot) including the division into 4 disjunct regions each.

Application of nonlinearity test to linear example

Table 1.1 summarizes the resultant values for the correlation coefficients, their confidence limits and the calculated upper and lower accuracy bound for each of the 4 disjunct regions. Table 1.2 shows the second eigenvalues of each of the correlation/covariance matrices for $h = 1$, 2, 3 and 4. Note that the values in italics correspond to the disjunct region for which the accuracy bounds have

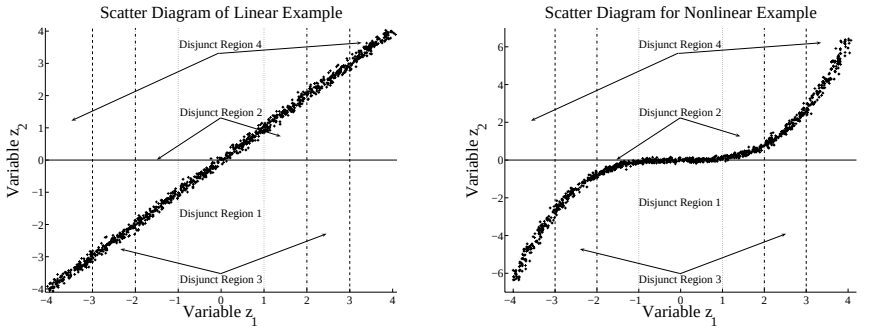


Fig. 1.2. Scatter diagrams for linear (left plot) and nonlinear simulation example (right plot) including boundaries for disjunct regions

Table 1.1. Correlation coefficients, their confidence limits and accuracy bounds for linear example

h	$r_{21}^{(h)}$	$r_{12_L}^{(h)}$	$r_{12_U}^{(h)}$	$\sigma_{h_{\min}}$	$\sigma_{h_{\min}}$
1	0.9852	0.9826	0.9874	0.0126	0.0174
2	0.9978	0.9975	0.9982	0.0018	0.0025
3	0.9992	0.9991	0.9993	0.0007	0.0009
4	0.9996	0.9995	0.9997	0.0003	0.0005

Table 1.2. Residual variances (second eigenvalues) for each combination of disjunct regions.

h/DisjunctRegion	1	2	3	4
1	*0.0148*	0.0143	0.0147	0.0142
2	0.0022	*0.0022*	0.0022	0.0021
3	0.0008	0.0008	*0.0008*	0.0007
4	0.0004	0.0004	0.0004	*0.0004*

been calculated. Figure 1.3 benchmarks these residual variances against the accuracy bounds for each of the disjunct regions. This comparison yields that no violations of the accuracy bounds arise, which, as expected, leads to the acceptance of the hypothesis that the underlying relationship between z_1 and z_2 is linear. Next, we investigate whether the nonlinearity test can reveal that the second example describes a nonlinear relationship between both variables.

Application of nonlinearity test to nonlinear example

Using the data generated by (1.27), Table 1.3 summarizes the resultant values for the correlation coefficients, their confidence limits and the calculated upper and lower accuracy bound for each of the 4 disjunct regions. Again, there is

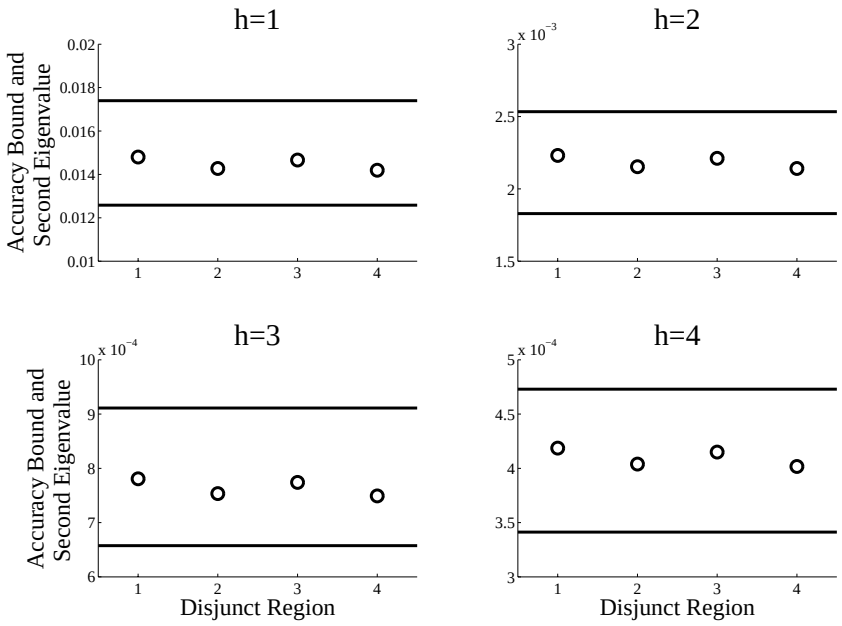


Fig. 1.3. Benchmarking of the residual variances against accuracy bounds of each disjunct region

Table 1.3. Correlation coefficients, their confidence limits and accuracy bounds for nonlinear example.

h	$r_{21}^{(h)}$	$r_{12_L}^{(h)}$	$r_{12_U}^{(h)}$	$\sigma_{h_{\min}}$	$\sigma_{h_{\min}}$
1	0.4341	0.3656	0.4979	0.5021	0.6344
2	0.9354	0.9244	0.9449	0.0551	0.0756
3	0.9752	0.9709	0.9789	0.0211	0.0291
4	0.9882	0.9861	0.9900	0.0100	0.0139

one principal component that describes the underlying relationship between z_1 and z_2. In contrast, the other component represents the superimposed noise sequences e_1 and e_2. However, this time, the underlying relationship is nonlinear.

Using the correlation/covariance matrices for each combination, Table 1.4 shows their second eigenvalue for $h = 1$, 2, 3 and 4. As before, the diagonal elements are marked in italics and represent the residual variance of the reconstructed data inside the disjunct region for which accuracy bounds have been computed and, by default, must fall inside these bounds. In contrast to the linear example, the residual variance of the reconstructed data for each

Table 1.4. Residual variances (second eigenvalues) for each combination of disjunct regions.

h/DisjunctRegion	1	2	3	4
1	*0.5659*	**0.7164**	**0.8866**	**0.8515**
2	**0.0258**	*0.0646*	**0.1145**	**0.1188**
3	**0.0017**	**0.0067**	*0.0248*	**0.0358**
4	**0.0002**	**0.0011**	**0.0056**	*0.0118*

of the other disjunct regions, fall outside the accuracy bounds. Note that the violating elements in Table 1.4 are marked in bold. This result is expected given the construction of the nonlinear data set. Figure 1.4 gives a graphical illustration for the results in Table 1.4.

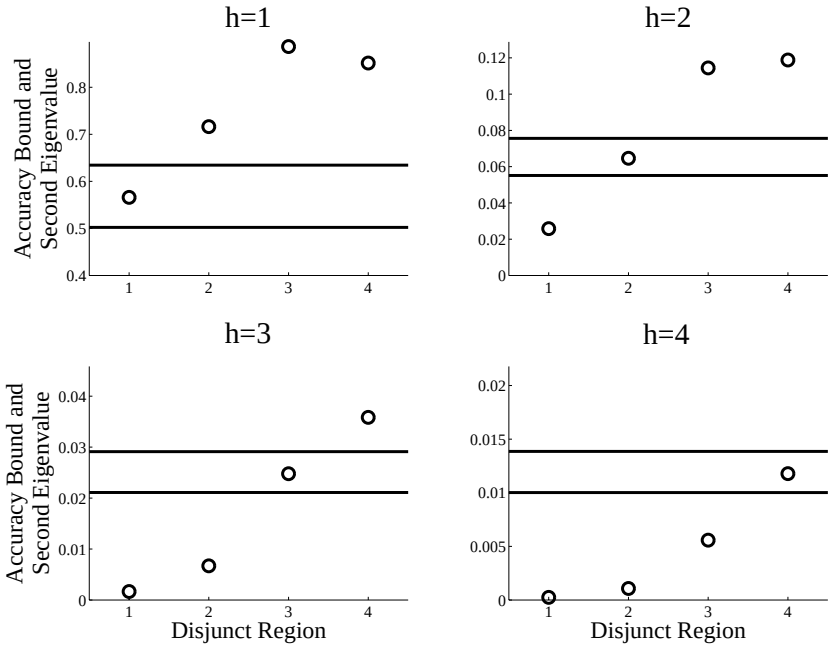


Fig. 1.4. Benchmarking of the residual variances against accuracy bounds of each disjunt region

1.4 Nonlinear PCA Extensions

This section reviews nonlinear PCA extensions that have been proposed over the past two decades. Hastie and Stuetzle [25] proposed *bending* the loading vectors to produce curves that approximate the nonlinear relationship between a set of two variables. Such curves, defined as *principal curves*, are discussed in the next subsection, including their multidimensional extensions to produce principal surfaces or *principal manifolds*.

Another paradigm, which has been proposed by Kramer [37], is related to the construction of an artificial neural network to represent a nonlinear version of (1.2). Such networks that map the variable set $\mathbf{z}$ to itself by defining a reduced dimensional bottleneck layer, describing nonlinear principal components, are defined as *autoassociative neural networks* and are revisited in Subsection 4.2.

A more recently proposed NLPCA technique relates to the definition of nonlinear mapping functions to define a feature space, where the variable space $\mathbf{z}$ is assumed to be a nonlinear transformation of this feature space. By carefully selecting these transformation using Kernel functions, such as radial basis functions, polynomial or sigmoid kernels, conceptually and computationally efficient NLPCA algorithms can be constructed. This approach, referred to as *Kernel PCA*, is reviewed in Subsection 4.3.

1.4.1 Principal Curves and Manifolds

A brief introduction into principal curves (PCs) is given next, followed by a geometric interpretation illustrating the progression from the PCA weight vector, associated to the largest eigenvalue of $\mathbf{S}_{ZZ}$ to a principal curve. The characteristics of principal curves are then outlined prior to algorithmic developments and refinements.

Introduction to principal curves

Principal Curves (PCs), presented by Hastie and Stuetzle [24, 25], are smooth one-dimensional curves passing through the *middle of a cloud representing a data set*. Utilizing probability distribution, a principal curve satisfies the self-consistent property, which implies that any point on the curve is the average of all data points projected onto it. As a nonlinear generalization of principal component analysis, PCs can be also regarded as a one-dimensional manifold embedded in high dimensional data space. In addition to the statistical property inherited from linear principal components, PCs also reflect the geometrical structure of data due. More precisely, the natural parameter *arc-length* is regarded as a projection index for each sample in a similar fashion to the score variable that represents the distance of the projected data point from the origin. In this respect, a one-dimensional nonlinear topological relationship between two variables can be estimated by a principal curve [85].

From a weight vector to a principal curve

Inherited from the basic paradigm of PCA, PCs assume that the intrinsic *middle structure* of data is a curve rather than a straight line. In relation to the total least squares concept [71], the cost function of PCA is to minimize the sum of projection distances from data points to a line. This produces the same solution as that presented in Section 2, Eq. (1.14). Geometrically, eigenvectors and their corresponding eigenvalues of $\mathbf{S}_{ZZ}$ reflect the principal directions and the variance along the principal directions of data, respectively. Applying the above analysis to the first principal component, the following properties can be established [5]:

1. Maximize the variance of the projection location of data in the principal directions.
2. Minimize the squared distance of the data points from their projections onto the 1st principal component.
3. Each point of the first principal component is the conditional mean of all data points projected into it.

Assuming the underlying interrelationships between the recorded variables are governed by:

$$\mathbf{z} = \mathbf{A}\mathbf{t} + \mathbf{e}\,, \tag{1.28}$$

where $\mathbf{z} \in \mathbb{R}^N$, $\mathbf{t} \in \mathbb{R}^n$ is the latent variable (or projection index for the PCs), $\mathbf{A} \in \mathbb{R}^{N\times n}$ is a matrix describing the linear interrelationships between data $\mathbf{z}$ and latent variables $\mathbf{t}$, and $\mathbf{e}$ represent statistically independent noise, i.e. $E\{\mathbf{e}\} = \mathbf{0}$, $E\{\mathbf{e}\mathbf{e}\} = \delta\mathbf{I}$, $E\left\{\mathbf{e}\mathbf{t}^T\right\} = \mathbf{0}$ with δ being the noise variance. PCA, in this context, uses the above principles of the first principal component to extract the n latent variables $\mathbf{t}$ from a recorded data set $\mathbf{Z}$.

Following from this linear analysis, a general nonlinear form of (1.28) is as follows:

$$\mathbf{z} = \mathbf{f}(\mathbf{t}) + \mathbf{e}\,, \tag{1.29}$$

where $\mathbf{f}(\mathbf{t})$ is a nonlinear function and represents the interrelationships between the latent variables $\mathbf{t}$ and the original data $\mathbf{z}$. Reducing $\mathbf{f}(\cdot)$ to be a linear function, Equation (1.29) clearly becomes (1.28), that is a special case of Equation (1.29).

To uncover the intrinsic latent variables, the following cost function, defined as

$$R = \sum_{i=1}^{K} \|\mathbf{z}_i - \mathbf{f}(\mathbf{t}_i)\|_2^2\,, \tag{1.30}$$

where K is the number available observations, can be used.

With respect to (1.30), linear PCA calculates a vector $\mathbf{p}_1$ for obtaining the largest projection index t_i of Equation (1.28), that is the diagonal elements of $E\left\{t^2\right\}$ represent a maximum. Given that $\mathbf{p}_1$ is of unit length, the location of

the projection of $\mathbf{z}_i$ onto the first principal direction is given by $\mathbf{p}_1 t_i$. Incorporating a total of n principal directions and utilizing (1.28), Equation (1.30) can be rewritten as follows:

$$R = \sum_{i=1}^{K} \|\mathbf{z}_i - \mathbf{P}\mathbf{t}_i\|_2^2 = \text{trace}\left\{\mathbf{Z}\mathbf{Z}^T - \mathbf{Z}^T\mathbf{A}\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\mathbf{A}\mathbf{Z}^T\right\} , \tag{1.31}$$

where trace$\{\cdot\}$ is the sum of the diagonal elements of matrix. Minimizing Equation (1.31) is equivalent to the determination of the largest eigenvalue of $\mathbf{Z}\mathbf{Z}^T$. Similarly, the distance function for PCs is defined as:

$$D^2\left(\mathbf{f}\left(\mathbf{t}\right)\right) = E\left\{\left(\mathbf{z} - \mathbf{f}\left(\mathbf{t}\right)\right)^T\left(\mathbf{z} - \mathbf{f}\left(\mathbf{t}\right)\right)\right\} , \tag{1.32}$$

where the variational approach is a main technique to minimize $D^2(\mathbf{f})$. It can be proven that the solution of Equation (1.32) is equivalent to that of Equation (1.31) if $\mathbf{f}\left(\cdot\right)$ reduces to a linear function. In the nonlinear case, it can be shown that the critical value for constructing the PC is the distance function.

Mathematically, let $\mathbf{f}$ be a PC and $\mathbf{g}$ be any curve of a family of curves $\mathbf{G}$ and define $\mathbf{f}_\epsilon = \mathbf{f}\left(\mathbf{t}\right) + \epsilon\mathbf{g}\left(\mathbf{t}\right)$ where ϵ is a distortion factor, $0 \leq \epsilon \leq 1$. The distance from the data $\mathbf{z}$ to $\mathbf{f}$ is defined as follows:

$$D^2(h, \mathbf{f}_\epsilon) = E_h \left\|\mathbf{z} - \mathbf{f}_\epsilon\left(\mathbf{t}_{\mathbf{f}_\epsilon}\left(\mathbf{z}\right)\right)\right\|_2^2 , \tag{1.33}$$

where $\mathbf{t}_{\mathbf{f}_\epsilon}$ is the projection index of the data point closest to the projection location on the curve $\mathbf{f}_\epsilon$ and $E_h(\cdot)$ is mathematical expectation of the given data distribution density h.

It can be proven that $\mathbf{f}$ is a critical value of the distance function in (1.33) under the assumption that Equation (1.34) is satisfied.

$$\left.\frac{dD^2\left(h, \mathbf{f}_\epsilon\right)}{d\epsilon}\right|_{\epsilon=0} \qquad \forall \mathbf{g} \in \mathbf{G} . \tag{1.34}$$

Therefore, the objective function (distance function) of PCs is a nonlinear generalization of PCA. The major difference is that the shapes of PCs are uncertain, whereas those of PCA are lines. Hence, it is necessary to address the differences as it is discussed below.

Characteristics of principal curves

To adhere to the properties of principal curves, some basic definitions of principal curves are given first.

Definition 2. *A one-dimensional curve embedded in* $\mathbb{R}^N$ *is a continuous function* $\mathbf{f} : \Xi \rightarrow \mathbb{R}^N$*, where* $\Xi = [a, b] \in \mathbb{R}$.

The curve $\mathbf{f}(\cdot)$ is a function that is parameterized by a single parameter $t \in \Xi$, that is $\mathbf{f}^T(t) = \left(\mathbf{f}_1(t) \cdots \mathbf{f}_n(t)\right)$, where $\mathbf{f}_1(t) \cdots \mathbf{f}_n(t)$ are referred to as coordinate functions.

Definition 3. *For any $\mathbf{z} \in \mathbb{R}^N$, the corresponding projection index $t_f(\mathbf{z})$ on the curve $\mathbf{f}(t)$ is defined as*

$$t_f(\mathbf{z}) = \sup\left\{t : \|\mathbf{z} - \mathbf{f}(t)\|_2^2 = \inf_{\tau} \|\mathbf{z} - \mathbf{f}(\tau)\|_2^2\right\}, \tag{1.35}$$

where $\mathbf{f}(t)$ is a curve in $\mathbb{R}^N$ parameterized by $t \in \mathbb{R}$.

The above definition can be interpreted as follows. The projection index $t_f(\mathbf{z})$ of $\mathbf{z}$ is the value of t for which $\mathbf{f}(t)$ is closest to $\mathbf{z}$. If there are multiple projection points that have an equal orthogonal distance to $\mathbf{z}$, the largest value of t is selected to remove ambiguity.

Hastie has proven that although ambiguous points may exist in the computation of PCs, the set of ambiguous points has a Lebesgue measure zero if the length of the curve is restricted. Hastie has further proven that for almost every $\mathbf{z}$, the projection function $t_{f_\epsilon}(\mathbf{z})$ is continuous under the compact support of probability density $\mathbf{h}$. The difference between 'continuous' and 'ambiguous' is: if $t_f(\mathbf{z})$ is continuous in $\mathbf{z}$, $\mathbf{z}$ is not an ambiguous point. The basic idea of projection index is illustrated in Figure 1.5.

Definition 4. *Based on the Definition 3, the distance between $\mathbf{z}$ and the curve $\mathbf{f}(t)$ is computed to be the squared distance between $\mathbf{z}$ and its projection point $\mathbf{f}(t_f(\mathbf{z}))$, that is:*

$$\Delta(\mathbf{z}, \mathbf{f}) = \|\mathbf{z} - \mathbf{f}(t_f(\mathbf{z}))\|_2^2 . \tag{1.36}$$

The projection distances from a data point to curve is an orthogonal distance rather than the vertical distances typically used by conventional regression methods.

Definition 5. *Given a curve $\mathbf{f}(t), t \in \mathbb{R}$, the arc-length, l, between t_0 and t_1 is given by:*

$$l = \int_{t_0}^{t_1} \|\mathbf{f}'(t)\|_2^2 \, \mathrm{d}t = \sum_{i=1}^{K-1} \|\mathbf{f}(t_{i+1}) - \mathbf{f}(t_i)\|_2^2 , \tag{1.37}$$

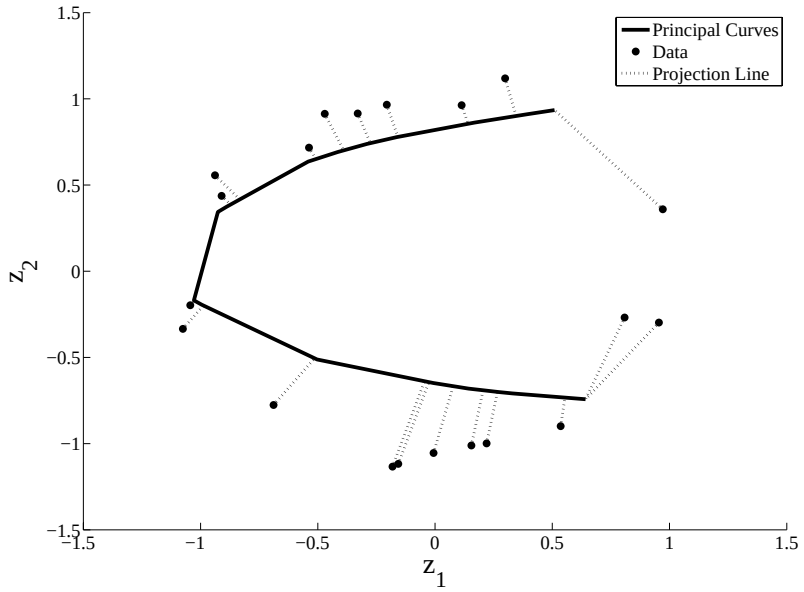
where $\mathbf{f}'(t)$ is tangent to the curve $\mathbf{f}$ at projection index t and is often described as the velocity vector.

If $\|\mathbf{f}'(t)\|_2^2 \equiv 1$ then $l = t_1 - t_0$ and such a curve is defined as a unit speed parameterized curve.

Definition 6. *The smooth curve $\mathbf{f}(\mathbf{t})$ is a principal curve if it:*

1. *does not intersect itself*
2. *has finite length inside any bounded subset of $\mathbb{R}^n$*

(a)



(b)

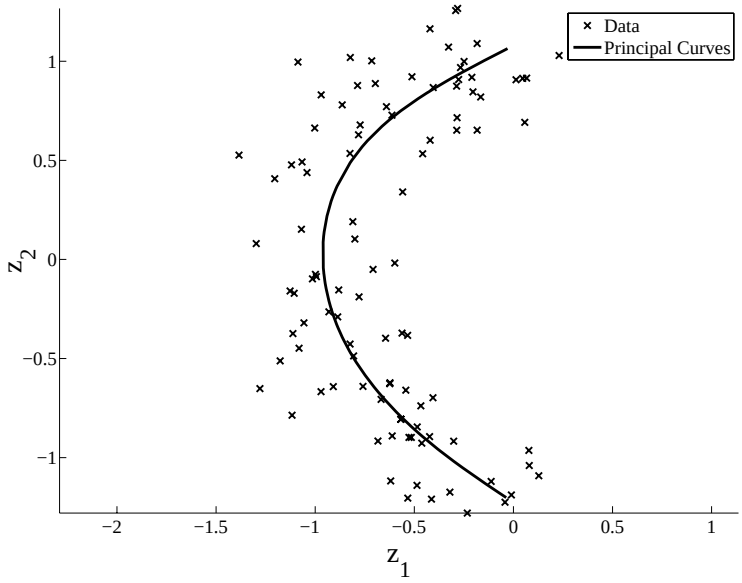


Fig. 1.5. Basic projection illustration from data point to curve (a) and a principal curve for a given data set (b)

3. is self-consistent, that is

$$\mathbf{f}(t) = E\{\mathbf{z}|\, t_f(\mathbf{z}) = t) \qquad \forall t \in \Xi\ . \tag{1.38}$$

The self-consistent property implies that each point on the curve is the conditional mean of the data points projected onto it. Thus, the principal curve is a smooth nonparametric self-consistent curve, which passes through the *middle of the distribution* and provides a one-dimensional nonlinear summary of the data.

Based on above definitions, Hastie and Stuetzle proposed a basic iterative algorithm for PCs, abbreviated as HSPCs for a given data distribution:

- Step1: Let the original curve $\mathbf{f}^{(0)}(t)$ be the first principal component where the subscript of $\mathbf{f}$ denotes the actual iteration number, commencing with j being equal to set zero.
- Step2 (projection step): $\forall \mathbf{z} \in \mathbb{R}^N$, compute:

$$t_{f^{(j)}}(\mathbf{z}) = \sup\left\{t : \left\|\mathbf{z} - \mathbf{f}^{(j)}(t)\right\|_2^2 = \inf_{\tau}\left\|\mathbf{z} - \mathbf{f}^{(j)}(\tau)\right\|_2^2\right\}\ . \tag{1.39}$$

- Step3 (expectation): According to self-consistency, recompute the curve $\mathbf{f}^{(j+1)}(t) = E\left\{\mathbf{Z}|\, t_{f^{(j)}}(\mathbf{Z}) = t\right\}$.
- Step4 (judgement): If $1 - \frac{\Delta(\mathbf{f}^{(j+1)})}{\Delta(\mathbf{f}^{(j)})} < \epsilon$, then stop, else set $j = j + 1$ and goto Step 2.

For the above iteration, ϵ is a predefined convergence criterion, which can be set to 0.01 for example.

If the data distribution is unknown, cubic smoothing splines can be used as an alternative strategy for the estimation of HSPCs. This entails finding $\mathbf{f}(t)$ and $t_i \in [0,1], i = \{1, \cdots, K\}$ so that

$$D^2(\mathbf{f}) = \sum_{i=1}^{K} \parallel \mathbf{z}_i - \mathbf{f}(t_i) \parallel_2^2 + \mu \int_0^1 \parallel \mathbf{f}''(t) \parallel_2^2 \, \mathrm{d}t \tag{1.40}$$

is minimized under the condition that the arc-length t is constrained to lie between $[0,1]$, μ is fixed smoothing factor, and $\mathbf{f}''(t)$ denotes the second-order derivative. More details of splines may be available in [60] for example.

Algorithmic developments

Since the concept was proposed by Hastie and Stuetzle in 1989, a considerable number of refinements and further developments have been reported. The first thrust of such developments address the issue of bias. The HSPCs algorithm has two biases, a *model bias* and an *estimation bias*.

Assuming that the data are subjected to some distribution function with gaussian noise, a *model bias* implies that that the radius of curvature in the

curves is larger than the actual one. Conversely, spline functions applied by the algorithm results in an estimated radius that becomes smaller than the actual one.

With regards to the *model bias*, Tibshirani [69] assumed that data are generated in two stages (i) the points on the curve $\mathbf{f}(t)$ are generated from some distribution function μ_t and (ii) $\mathbf{z}$ are formed based on conditional distribution $\mu_{z|t}$ (here the mean of $\mu_{z|t}$ is $\mathbf{f}(t)$). Assume that the distribution functions μ_t and $\mu_{z|t}$ are consistent with μ_z, that is $\mu_z = \int \mu_{z|t}(\mathbf{z}|t)\,\mu_t(t)\,\mathrm{d}t$. Therefore, $\mathbf{z}$ are random vectors of dimension N and subject to some density μ_z. While the algorithm by Tibshirani [69] overcomes the *model bias*, the reported experimental results in this paper demonstrate that the practical improvement is marginal. Moreover, the self-consistent property is no longer valid.

In 1992, Banfield and Raftery [4] addressed the *estimation bias* problem by replacing the squared distance error with residual and generalized the PCs into closed-shape curves. However, the refinement also introduces numerical instability and may form a smooth but otherwise incorrect principal curve.

In the mid 1990s, Duchamp and Stuezle [18, 19] studied the holistical differential geometrical property of HSPCs, and analyzed the first and second variation of principal curves and the relationship between self-consistent and curvature of curves . This work discussed the existence of principal curves in the sphere, ellipse and annulus based on the geometrical characters of HSPCs. The work by Duchamp and Stuezle further proved that under the condition that curvature is not equal to zero, the expected square distance from data to principal curve in the plane is just a saddle point but not a local minimum unless low-frequency variation is considered to be described by a constraining term. As a result, cross-validation techniques can not be viewed as an effective measure to be used for the model selection of principal curves.

At the end of the 1990s, Kégl proposed a new principal curve algorithm that incorporates a length constraint by combining vector quantization with principal curves. For this algorithm, further referred to as the KPC algorithm, Kégl proved that if and only if the data distribution has a finite second-order moment, a KPC exists and is unique. This has been studied in detail based on the principle of structural risk minimization, estimation error and approximation error. It is proven in references [34, 35] that the KPC algorithm has a faster convergence rate that the other algorithms described above. This supports to use of the KPC algorithm for large databases.

While the KPCs algorithm presents a significant improvement, several problems still remain. For example, the first principal component is often assumed to be an initial segment for the KPCs algorithm. For complex data which are subject to an uneven and/or sparse distribution, however, a good estimate of the initial curve plays a crucial role in order to guarantee that the algorithm converges to the actual principal curve. Secondly, the computational complexity gradually rises with an increase in the number of segments. However, if some vertices to be optimized go outside the domain of data, the algorithm has no ability to detect and remove this so that the subsequent

optimization and projection steps may fail. Thirdly, many parameters need to be predetermined and adjusted based on heuristic experience, which may hamper the practical usefulness of the KPC algorithm.

Addressing these drawbacks, Zhang and Chen [83] proposed a *constraint K-Segment principal curve* or CKPC algorithm. For this algorithm, the initial and final points of the curve are predefined by introducing data from the unobservable region or prior knowledge so that a better initial curves can be extracted. Secondly, a new constrained term for the removal of some abnormal vertices is presented to prevent subsequent optimization and projection steps to fail. Experiments involving intelligent transportation systems demonstrated that the CKPC algorithm provides a stronger generalization property than the KPC algorithm [83].

Morales [46] stated that from a differential manifold viewpoint a principal curves is a special case of manifold fitting. Morales work further generalized principal curves into principal embedding and introduced harmonic energy to be a regularizing term for determining a local minimum of the principal embedding. However, this work does not provide a practical algorithm for constructing a principal embedding. However, Smola [61] pointed out that most of the unsupervised learning approaches, such as principal curves, can rely on vector quantization, and proposed regularized principal manifold or RPM approach. Smola further proved the equivalence of the approach with the KPC algorithm, and derived a consistent convergence bound based on statistical learning theory.

Delicado [14] reconsidered the relation between principal curves and linear principal component analysis and introduced the concept of principal curves of oriented points or PCOP. This analysis was motivated by the fact that the first principal component goes through the conditional mean in a hyperplane and is orthogonal to the hyperplane which minimizes conditional total variance. When repeated searching from different samples, multiple points which satisfy the property of conditional mean value can be found. These points are called PCOP and the principal curve is the one across the PCOP. Similarly, the total-variance property can be recursively generalized to higher-order continuous principal curves. The drawback of this approach, however, is its considerable computational complexity.

Alternative approaches to principal curves include the work by Chang and Ghosh [7, 8], who define probabilistic principal curves, PPCs, and probabilistic principal surfaces, PPSs based on a generative topography mapping [6]. Different from the other approaches discussed above, PPCs and PPSs assume that high-dimensional topographical relationships among data are explicitly generated from some low-dimensional latent variables, and the algorithms are devoted to calculating the probability density of given data sets based on their latent variable structure. Advantages of these algorithms are that they not only keep the self-consistency, but also generalize principal curves into 2 dimensional to 3 dimensional principal manifolds through the introduction of parameterized models.

Verbeek [73] proposed a soft K-segments principal curves (SKPCs) algorithm, which first constructs K segments where each segment is obtained by local principal component analysis. These segments are then connected by minimizing the total degree of smoothness. A potential drawback of the SKPC algorithm is that it cannot be generalized into a high-dimension surface and the cost function does not consider a curvature penalty. Sandilya and Kulkarni [55] presented principal curves with bounded turn (PCBT). In a similar fashion to the KPC algorithm, PCBTs exists if and only if the data has a finite second-order moment.

Practically, principal curve algorithms have gained attention in a variety of applications, such as the alignment of magnets of the Stanford linear collier, identifying profiles of ice floes in the satellite images, handwriting ossification, speech recognition etc. [85, 53, 54, 26]. More recently, the principal curves algorithm by Hastie and Stuetzle [25] has also been applied to the modeling of freeway traffic streams [9] and the learning of high-precision GPS data from low-precision counterpart [84].

1.4.2 Neural Network Approaches

Using the structure shown in Figure 1.6, Kramer [37] proposed an alternative NLPCA implementation to principal curves and manifolds. This structure

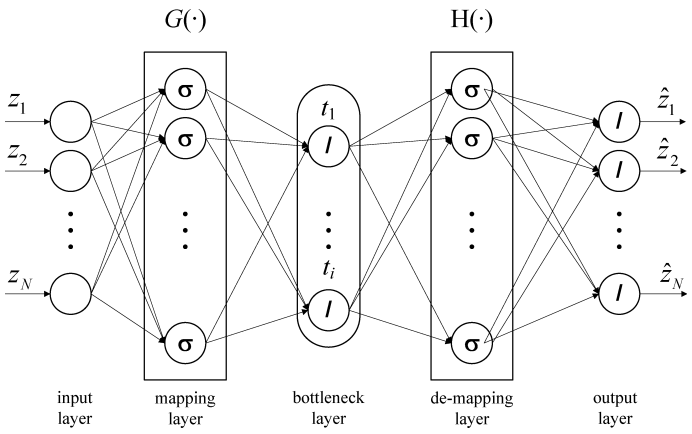


Fig. 1.6. Topology of autoassociative neural networks.

represents an autoassociative neural network (ANN), which, in essence, is an identify mapping that consists of a total of 5 layers. Identify mapping relates to this network topology is optimized to reconstruct the N network input variables as accurately as possible using a reduced set of bottleneck nodes $n < N$. From the left to right, the first layer of the ANN is the *input layer* that passes weighted values of the original variable set $\mathbf{z}$ onto the second layer,

that is the *mapping layer*:

$$\xi_i = \sum_{j=1}^{N} w_{ij}^{(1)} z_j + b_i^1 \ , \tag{1.41}$$

where $w_{ij}^{(1)}$ are the weights for the first layer and $b_i^{(1)}$ is a bias term. The sum in (1.41), ξ_i, is the input the the ith node in the mapping layer that consists of a total of M_m nodes. A scaled sum of the outputs of the nonlinearly transformed values $\sigma(\xi_i)$, then produce the nonlinear scores in the bottleneck layer. More precisely, the pth nonlinear score t_p, $1 \leq p \leq n$ is given by:

$$t_p = \sum_{i=1}^{M_m} w_{pi}^{(2)} \sigma(\xi_i) + b_p^{(2)} = \sum_{i=1}^{M_m} w_{pi}^{(2)} \sigma\left(\sum_{j=1}^{N} w_{ij}^{(1)} z_j + b_i^1 \right) + b_p^{(2)} \ . \tag{1.42}$$

To improve the modeling capability of the ANN structure for mildly nonlinear systems, it is useful to include linear contributions of the original variables $z_1 \ z_2 \cdots z_N$:

$$t_p = \sum_{i=1}^{M_m} w_{pi}^{(2)} \sigma\left(\sum_{j=1}^{N} w_{ij}^{(1)} z_j + b_i^1 \right) + \sum_{j=1}^{N} w_{pi}^{(1l)} z_i + b_p^{(2)} \ , \tag{1.43}$$

where the index l refers to the linear contribution of the original variables. Such a network, where a direct linear contribution of the original variables is included, is often referred to as a *generalized neural network*. The middle layer of the ANN topology is further referred to as the *bottleneck layer*.

A linear combination of these nonlinear score variables then produces the inputs for the nodes in the 4th layer, that is the *demapping layer*:

$$\tau_j = \sum_{p=1}^{n} w_{jp}^{(3)} t_p + b_p^{(3)} \ . \tag{1.44}$$

Here, $w_{jp}^{(3)}$ and $b_p^{(3)}$ are the weights and the bias term associated with the bottleneck layer, respectively, that represents the input for the jth node of the demapping layer. The nonlinear transformation of τ_j finally provides the reconstruction of the original variables $\mathbf{z}$, $\widehat{\mathbf{z}} = \left(\widehat{z}_1 \ \widehat{z}_2 \cdots \widehat{z}_N \right)^T$ by the *output layer*:

$$\widehat{z}_q = \sum_{j=1}^{M_d} w_{qj}^{(4)} \sigma\left(\sum_{p=1}^{n} w_{jp}^{(3)} t_p + b_p^{(3)} \right) + \sum_{j=1}^{n} w_{qj}^{(3l)} t_j + b_q^{(4)} \ , \tag{1.45}$$

which may also include a linear contribution of the nonlinear score variables, indicated by the inclusion of the term $\sum_{j=1}^{n} w_{qj}^{(3l)} t_j$. Usually, the training of the

network weights and bias terms is done using a gradient descent approach like the computationally efficient Levenberg-Marquardt algorithm. It should be noted that the functional relationships between the original variable set $\mathbf{z} \in \mathbb{R}^N$ and the nonlinear score variables $\mathbf{t} \in \mathbb{R}^n$ is further referred to as the mapping function $\mathbf{G}(\cdot)$. Furthermore, the functional relationship between the nonlinear score variables and the reconstructed original variables $\widehat{\mathbf{z}} \in \mathbb{R}^N$ is defined as the demapping function $\mathbf{H}(\cdot)$.

To symbolize a nonlinear version of the iterative Power method for computing linear PCA, Kramer [37] proposed the following network topology, shown in Figure 1.7. In a close resemblance to the total least squares prob-

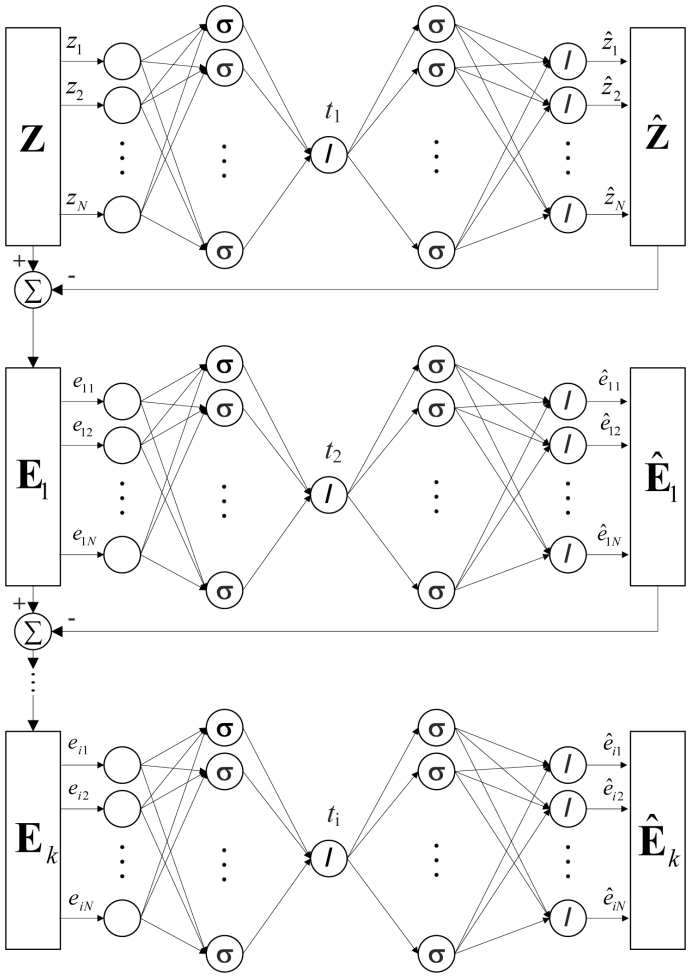


Fig. 1.7. Topology of a sequential autoassociative neural network.

lem [71] and the properties for first few principal components by Barnett [5], summarized in Subsection 4.1, this sequential ANN approach minimizes the squared distance of the data points from their reconstructions using the first nonlinear principal component t_1. After deflating the data matrix $\mathbf{E}_1 = \mathbf{Z} - \widehat{\mathbf{Z}}$, the second nonlinear principal component t_2 is again obtained such that the squared distance of the of the residuals, stored in $\mathbf{E}_1$ and the reconstruction of $\mathbf{E}_1$, $\widehat{\mathbf{E}}_2$ is minimized. This iteration continues until the residual variance of $\mathbf{E}_k$ is sufficiently small.

Tan and Mavrovouniotis [68] pointed out that the ANN topology is complex and difficult to train. As a rule of thumb, an increase in the number of hidden layers produces a deterioration in the performance of backpropagation based algorithms, such as the Levenberg-Marquardt one [27]. The article by Tan and Mavrovouniotis also presents a simplified application study describing a circle which demonstrates the difficulties of training such a large network topology. To overcome this deficiency, they proposed an *input training* network that has the topology shown in Figure 1.8 for a total of 5 original variables, $z_1\ z_2\ z_3\ z_4\ z_5$ and two nonlinear principal components $t_1\ t_2$. The

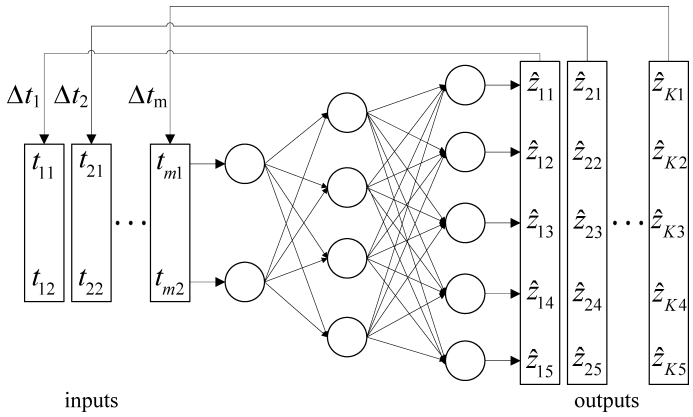


Fig. 1.8. Topology of the input training network.

input training or IT network determines the network weights as well as the score variables in an iterative fashion. For a minimum of the squared distance between the original and reconstructed observations, the following cost function can be used:

$$J = \sum_{i=1}^{K} \sum_{j=1}^{N} \left(z_{ij} - \widehat{z}_{ij}\right)^2 = \sum_{i=1}^{K} \sum_{j=1}^{N} \left(z_{ij} - \theta_j\left(\mathbf{t}_i\right)\right)^2 \ , \tag{1.46}$$

where $\theta_j(\cdot)$ is the network function for reconstructing the jth original variable z_j and $\mathbf{t}_i^T = \left(t_1\ t_2 \cdots t_n\right)$. Under the assumption that the network weights are constant and predetermined, the *steepest descent* in the direction of optimal

network inputs is given by the gradient of J with respect to the nonlinear scores:

$$\Delta t_{ik} = -\frac{\partial J}{\partial t_{ik}} = \sum_{j=1}^{N} 2\left(z_{ij} - \theta_j\left(\mathbf{t}_i\right)\right) \frac{\partial \theta_j\left(\mathbf{t}_i\right)}{\partial t_{ik}} \ . \tag{1.47}$$

Reconstructing z_{ij}, that is determining $\widehat{z}_{ij}$ using the IT network, is based on the input and middle layers:

$$\widehat{z}_{ij} = \theta_j\left(\mathbf{t}_i\right) = \sum_{p=1}^{M_d} w_{pj}^{(2)} \sigma\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_p\right) \ , \tag{1.48}$$

where the indices (1) and (2) refer to the input and middle layer weights and b_p is a bias term. For simplicity, linear terms are not included, which, however, does not restrict generality. Combining Equations (1.47) and (1.48) gives rise to determine the steepest descent in the direction of the network training inputs between the input and hidden layer:

$$\Delta t_{ik} = \sum_{p=1}^{N} w_{pj}^{(1)} \delta_{ip} \ , \tag{1.49}$$

where:

$$\delta_{ip} = \sum_{j=1}^{N} 2\left(z_{ij} - \theta_j\left(\mathbf{t}_i\right)\right) \frac{\partial \theta_j\left(\mathbf{t}_i\right)}{\partial t_{ik}} \tag{1.50}$$

which is given by:

$$\frac{\partial \theta_j\left(\mathbf{t}_i\right)}{\partial t_{ik}} = \sum_{p=1}^{M_d} w_{pj}^{(2)} \frac{\partial}{\partial t_{ik}} \sigma\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_p\right) \ , \tag{1.51}$$

and, hence,

$$\frac{\partial \theta_j\left(\mathbf{t}_i\right)}{\partial t_{ik}} = w_{kj}^{(2)} \sigma'\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_q\right) \ , \tag{1.52}$$

so that δ_{ip} can finally be determined as

$$\delta_{ip} = \sigma'\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_q\right) \sum_{j=1}^{N} 2 w_{kj}^{(2)} \left(z_{ij} - \theta_j\left(\mathbf{t}_i\right)\right) \ . \tag{1.53}$$

Following the above derivative, the steepest descent direction for the training network weights between the input and hidden layer can be obtained as follows:

$$\Delta w_{pk}^{(1)} = -\frac{\partial J}{\partial w_{pk}^{(1)}} = \sum_{i=1}^{K} \sum_{j=1}^{N} \frac{\partial}{\partial w_{pk}^{(1)}} \left(z_{ij} - \sum_{p=1}^{M_d} w_{pj}^{(2)} \sigma\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_p\right)\right)^2 \ , \tag{1.54}$$

which is given by

$$\Delta w_{pk}^{(1)} = \sum_{i=1}^{K} \sum_{j=1}^{N} 2\left(z_{ij} - \sum_{p=1}^{M_d} w_{pj}^{(2)} \sigma\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_p\right)\right) t_{ik} \sigma'\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_p\right) , \tag{1.55}$$

which in a simplified form is given by

$$\Delta w_{pk}^{(1)} = \sum_{i=1}^{K} \eta_{ik}^{(1)} \sum_{j=1}^{N} \left(z_{ij} - \sum_{p=1}^{M_d} w_{pj}^{(2)} \sigma\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_p\right)\right) , \tag{1.56}$$

where $\eta_{ik}^{(1)} = t_{ik} \sigma'\left(\sum_{q=1}^{n} w_{pq}^{(1)} t_{iq} + b_p\right)$. In a similar fashion to (1.55), a steepest descent can also be derived for the network weights $w_{pk}^{(2)}$. Using the steepest descents, derived above, the IT network can be trained using backpropagation techniques.

Dong and McAvoy [16] proposed another approach to simplify the structure of the original complex 5 layer structure by Kramer. This work relies on a separation of the 5 layer network into the 3 layer mapping function $\mathbf{G}(\cdot)$ and another 3 layer network representing the demapping function $\mathbf{H}(\cdot)$. According to Figure 1.6, the input of the mapping function are the original variables $\mathbf{z} \in \mathbb{R}^N$ and the output are the nonlinear score variables $\mathbf{t} \in \mathbb{R}^n$, while the inputs and outputs of the demapping function are the score variables and reconstructed variables $\widehat{\mathbf{z}} \in \mathbb{R}^N$, respectively. Instead of determining the inputs to the demapping function $\mathbf{H}(\cdot)$ by optimizing Equation (1.46) with respect to the nonlinear score variables for each observation, the approach by Dong and McAvoy utilizes principal curves to determine nonlinear score variables.

A potential problem of this approach has been discussed by Jia *et al.* [31]. The principal curve approach by Dong and McAvoy assumes that the approximation of a nonlinear function can be achieved by a linear combination of a number of univariate nonlinear functions. This, however, is a restriction of generality and implies that only a limited class of nonlinear functions can be approximated using this technique. In contrast, the algorithm by Tan and Mavrovouniotis does not suffer from this inherent limitation.

1.4.3 Kernel PCA

This subsection details the principles of kernel PCA, which has been proposed by Schölkopf *et al.* [57]. An introduction into the principles of kernel PCA, including the definition of the covariance matrix in the feature space, is given next, followed by an outline of how to compute a kernel PCA model using the kernel matrix. It is finally shown how to extract the score variables from the kernel PCA model.

Introduction to kernel PCA

This technique first maps the original input vectors $\mathbf{z}$ onto a high-dimensional feature space $\mathbf{z} \mapsto \mathbf{\Phi}(\mathbf{z})$ and then perform the principal component analysis on $\mathbf{\Phi}(\mathbf{z})$. Given a set of observations $\mathbf{z}_i \in \mathbb{R}^N$, $i = \left\{ 1\ 2 \cdots K \right\}$, the mapping of $\mathbf{z}_i$ onto a feature space, that is $\mathbf{\Phi}\left(\mathbf{z}\right)$ whose dimension is considerably larger than N, produces the following sample covariance matrix:

$$\mathbf{S}_{\Phi\Phi} = \frac{1}{K-1}\sum_{i=1}^{K}\left(\mathbf{\Phi}\left(\mathbf{z}_i\right) - \mathbf{m}_{\Phi}\right)\left(\mathbf{\Phi}\left(\mathbf{z}_i\right) - \mathbf{m}_{\Phi}\right)^T = \frac{1}{K-1}\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)^T\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)\ . \tag{1.57}$$

Here, $\mathbf{m}_{\Phi} = \frac{1}{K}\mathbf{\Phi}\left(\mathbf{Z}\right)^T\mathbf{1}_K$, where $\mathbf{1}_K \in \mathbb{R}^K$ is a column vector storing unity elements, is the sample mean in the feature space, and $\mathbf{\Phi}\left(\mathbf{Z}\right) = \left[\ \mathbf{\Phi}\left(\mathbf{z}_1\right)\ \mathbf{\Phi}\left(\mathbf{z}_2\right)\cdots\ \ \mathbf{\Phi}\left(\mathbf{z}_K\right)\ \right]^T$ and $\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right) = \mathbf{\Phi}\left(\mathbf{Z}\right) - \frac{1}{K}\mathbf{E}_K\mathbf{\Phi}(\mathbf{Z})$, with $\mathbf{E}_K$ being a matrix of ones, are the original and mean centered feature matrices, respectively.

KPCA now solves the following eigenvector-eigenvalue problem,

$$\mathbf{S}_{\Phi\Phi}\mathbf{p}_i = \frac{1}{K-1}\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)^T\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)\mathbf{p}_i = \lambda_i\mathbf{p}_i \qquad i = 1, 2\cdots N\ , \tag{1.58}$$

where λ_i and $\mathbf{p}_i$ are the eigenvalue and its associated eigenvector of $\mathbf{S}_{\Phi\Phi}$, respectively. Given that the explicit mapping formulation of $\mathbf{\Phi}(\mathbf{z})$ is usually unknown, it is difficult to extract the eigenvector-eigenvalue decomposition of $\mathbf{S}_{\Phi\Phi}$ directly. However, KPCA overcomes this deficiency as shown below.

Determining a kernel PCA model

Starting from the eigenvector-eigenvalue decomposition of $\mathbf{G} = \bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)^T$, which is further defined as the Gram matrix:

$$\bar{\mathbf{\Phi}}(\mathbf{Z})\bar{\mathbf{\Phi}}(\mathbf{Z})^T\mathbf{v}_i = \zeta_i\mathbf{v}_i\ , \tag{1.59}$$

where ζ_i and $\mathbf{v}_i$ are the eigenvalue and its eigenvector, respectively, carrying out a pre-multiplication of (1.59) by $\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)^T$ produces:

$$\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)^T\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)^T\mathbf{v}_i = \zeta_i\bar{\mathbf{\Phi}}\left(\mathbf{Z}\right)^T\mathbf{v}_i \qquad i = 1\ 2\cdots N\ . \tag{1.60}$$

By comparing (1.60) and (1.58), it now follows that

$$\zeta_i/(K-1)\ \text{ and }\ \bar{\mathbf{\Phi}}(\mathbf{Z})^T\mathbf{v}_i/\left\|\bar{\mathbf{\Phi}}(\mathbf{Z})^T\mathbf{v}_i\right\|_2$$

are also corresponding eigenvalues and eigenvectors of $\mathbf{S}_{\Phi\Phi}$, that is:

$$\begin{aligned}&\lambda_i = \zeta_i/(K-1),\\ &\mathbf{p}_i = \bar{\mathbf{\Phi}}(\mathbf{Z})^T\mathbf{v}_i\Big/\sqrt{\mathbf{v}_i^T\bar{\mathbf{\Phi}}(\mathbf{Z})\bar{\mathbf{\Phi}}(\mathbf{Z})^T\mathbf{v}_i} = \bar{\mathbf{\Phi}}(\mathbf{Z})^T\mathbf{v}_i/\sqrt{\zeta_i}\ .\end{aligned} \tag{1.61}$$

By defining a kernel function $\psi(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{\Phi}(\mathbf{z}_i)^T \mathbf{\Phi}(\mathbf{z}_j)$, the Gram matrix $\mathbf{G}$ can be constructed from a *kernel matrix* $\mathbf{\Psi}(\mathbf{Z}) \in \mathbb{R}^{K \times K}$ whose elements ψ_{ij} are $\psi(\mathbf{z}_i, \mathbf{z}_j)$,

$$\mathbf{G} = \mathbf{\Psi}(\mathbf{Z}) - \frac{1}{K}\mathbf{\Psi}(\mathbf{Z})\,\mathbf{E}_K - \frac{1}{K}\mathbf{E}_K\mathbf{\Psi}(\mathbf{Z}) + \frac{1}{K^2}\mathbf{E}_K\mathbf{\Psi}(\mathbf{Z})\,\mathbf{E}_K\,. \tag{1.62}$$

It is important to note that the calculation of $\mathbf{G}$ (i) only requires the kernel formulation of $\psi(\mathbf{z}_i, \mathbf{z}_j)$ and (ii) but no *a priori* knowledge of the exact mapping $\mathbf{\Phi}(\mathbf{z})$. The most commonly used kernel functions include polynomial, RBF and Sigmoid kernels [59].

Calculation of the score variables

Assuming that a PCA model has been constructed from the covariance matrix $\mathbf{S}_{\Phi\Phi}$, that is $\mathbf{S}_{\Phi\Phi} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$, incorporating Equation (1.61) gives rise to:

$$\mathbf{P} = \left[\,\mathbf{p}_1\;\mathbf{p}_2 \cdots \mathbf{p}_n\,\right] = \bar{\mathbf{\Phi}}(\mathbf{Z})^T \left[\,\mathbf{v}_1/\sqrt{\zeta_1}\;\mathbf{v}_2/\sqrt{\zeta_2} \cdots \mathbf{v}_n/\sqrt{\zeta_n}\,\right]\,. \tag{1.63}$$

Redefining $\bar{\mathbf{\Phi}}(\mathbf{Z})$, as shown in (1.57), and rewriting (1.63) produces:

$$\mathbf{P} = \left[\mathbf{\Phi}(\mathbf{Z})^T - \frac{1}{K}\mathbf{\Phi}(\mathbf{Z})^T\,\mathbf{E}_N\right]\mathbf{V} = \mathbf{\Phi}(\mathbf{Z})^T\,\mathbf{A}\,, \tag{1.64}$$

where $\mathbf{V} = \left[\mathbf{v}_1/\sqrt{\zeta_1}\;\mathbf{v}_2/\sqrt{\zeta_2} \cdots \mathbf{v}_n/\sqrt{\zeta_n}\right]$, $\mathbf{A} = \left[\mathbf{I}_K - \frac{1}{K}\mathbf{E}_K\right]\mathbf{V}$ with $\mathbf{I}_K$ being the identify matrix of dimension K. Utilizing Equation (1.64), the score variables, stored in the vector $\mathbf{t}$, can now be obtained as follows:

$$\mathbf{t} = \mathbf{P}^T\left[\mathbf{\Phi}(\mathbf{z}) - \mathbf{m}_\Phi\right] = \mathbf{A}^T\mathbf{\Phi}(\mathbf{Z})\left[\mathbf{\Phi}(\mathbf{z}) - \frac{1}{K}\mathbf{\Phi}(\mathbf{Z})\,,\mathbf{1}_K\right] \tag{1.65}$$

which, using the definition of the kernel function $\psi(\cdot)$, can finally be written as shown below:

$$\mathbf{t} = \mathbf{A}^T\left(\boldsymbol{\psi}(\mathbf{Z}, \mathbf{z}) - \frac{1}{K}\mathbf{\Psi}(\mathbf{Z})\,\mathbf{1}_K\right)\,. \tag{1.66}$$

Here, $\boldsymbol{\psi}(\mathbf{Z}, \mathbf{z})$ is the *kernel vector* for the new observation $\mathbf{z}$ based on the set of reference observations $\mathbf{Z}$, that is $\boldsymbol{\psi}(\mathbf{Z}, \mathbf{z}) = \left(\,\psi(\mathbf{z}_1, \mathbf{z})\;\psi(\mathbf{z}_2, \mathbf{z}) \cdots \psi(\mathbf{z}_K, \mathbf{z})\,\right)^T$.

1.5 Analysis of Existing Work

This section provides a comparison between each of the proposed nonlinear approaches in terms of their computational demand and their ability to represent a generalization of linear PCA. The section finally investigates potential research areas that have not been addressed and require require further attention.

1.5.1 Computational Issues

This subsection investigates computationally related issues for principal curves and manifolds first, followed by the analysis of neural network techniques and finally the Kernel PCA approach.

Principal curve and manifold approaches

Resulting from the fact that the nearest projection coordinate of each sample in the curve is searched along the whole line segments, the computational complexity of the HSPCs algorithm is of order $O(n^2)$[25] which is dominated by the projection step. The HSPCs algorithm, as well as other algorithms proposed by [69, 4, 18, 19], may therefore be computationally expensive for large data sets.

For addressing the computational issue, several strategies are proposed in subsequently refinements. In reference [8], the PPS algorithm supposes that the data are generated from a collection of latent nodes in low-dimensional space, and the computation to determine the projections is achieved by comparing the distances among data and the high-dimensional counterparts in the latent nodes. This results in a considerable reduction in the computational complexity if the number of the latent nodes is less than that number of observations. However, the PPS algorithm requires additional $O(N^2n)$ operations (Where n is the dimension of latent space) to compute an orthonormalization. Hence, this algorithm is difficult to generalize in high-dimensional spaces.

In [73], local principal component analysis in each neighborhood is employed for searching a local segment. Therefore, the computational complexity is closely relate to the number of local PCA models. However, it is difficulty for general data to combine the segments into a principal curve because a large number of computational steps are involved in this combination.

For the work by Kégl [34, 35], the KPCs algorithm is proposed by combining the vector quantization with principal curves. Under the assumption that data have finite second moment, the computational complexity of the KPCs algorithm is $O(n^{5/3})$ which is slightly less than that of the HSPCs algorithm. When allowing to add more than one vertex at a time, the complexity can be significantly decreased. Furthermore, a speed-up strategy discussed by Kégl [33] is employed for the assignments of projection indices for the data during the iterative projection procedure of the ACKPCs algorithms. If $\delta v^{(j)}$ is the maximum shift of a vertex v_j in the jth optimization step defined by:

$$\delta v^{(j)} = \max_{i=1,\cdots,k+1} \|v_i^{(j)} - v_i^{(j+1)}\|,$$

then after the $(j + j_1)$ optimization step, s_{i_1} is still the nearest line segment to x if

$$d(x, s_{i_1}^{(j)}) \leq d(x, s_{i_2}^{(j)}) - 2 \sum_{l=j}^{j+j_1} \delta v^{(l)} \ . \tag{1.67}$$

Further reference to this issue may be found in [33], pp. 66-68. Also, the stability of the algorithm is enhanced while the complexity is the equivalent to that of the KPCs algorithm.

Neural network approaches

The discussion in Subsection 4.2 highlighted that neural network approaches to determine a NLPCA model are difficult to train, particulary the 5 layer network by Kramer [37]. More precisely, the network complexity increases considerably if the number of original variables $\mathbf{z}$, N, rises. On the other hand, an increasing number of observations also contribute to a drastic increase in the computational cost. Since most of the training algorithms are iterative in nature and employ techniques based on the backpropagation principle, for example the Levenberg-Marquardt algorithm for which the Jacobian matrix is updated using backpropagation, the performance of the identified network depends on the initial selection for the network weights. More precisely, it may be difficult to determine a minimum for the associated cost function, that is the sum of the minimum distances between the original observations and the reconstructed ones.

The use of the IT network [68] and the approach by Dong and McAvoy [16], however, provide considerably simpler network topologies that are accordingly easier to train. Jia *et al.* [31] argued that the IT network can generically represent smooth nonlinear functions and raised concern about the technique by Dong and McAvoy in terms of its flexibility in providing generic nonlinear functions. This concern related to to concept of incorporating a linear combination of nonlinear function to estimate the nonlinear interrelationships between the recorded observations. It should be noted, however, that the IT network structure relies on the condition that an functional injective relationship exit between the score variables and the original variables, that is a unique mapping between the scores and the observations exist. Otherwise, the optimization step to determine the scores from the observations using the identified IT network may converge to different sets of score values depending on the initial guess, which is undesirable. In contrast, the technique by Dong and McAvoy does not suffer from this problem.

Kernel PCA

In comparison to neural network approaches, the computational demand for a KPCA insignificantly increase for larger values of N, size of the original variables set $\mathbf{z}$, which follows from (1.59). In contrast, the size of the Gram matrix increases quadratically with a rise in the number of analyzed observations, K. However, the application of the numerically stable singular value decomposition to obtain the eigenvalues and eigenvectors of the Gram matrix does not present the same computational problems as those reported for the neural network approaches above.

1.5.2 Generalization of Linear PCA?

The generalization properties of NLPCA techniques is first investigated for neural network techniques, followed for principal curve techniques and finally kernel PCA. Prior to this analysis, however, we revisit the cost function for determining the kth pair of the score and loading vectors for linear PCA. This analysis is motivated by the fact that neural network approaches as well as principal curves and manifolds minimize the residual variances. Reformulating Equations (1.9) and (1.10) to minimize the residual variance for linear PCA gives rise to:

$$\mathbf{e}_k = \mathbf{z} - t_k \mathbf{p}_k \ , \tag{1.68}$$

which is equal to:

$$J_k = E\left\{\mathbf{e}_k^T \mathbf{e}_k\right\} = E\left\{\left(\mathbf{z} - t_k \mathbf{p}_k\right)^T \left(\mathbf{z} - t_k \mathbf{p}_k\right)\right\} \ , \tag{1.69}$$

and subject to the following constraints

$$t_k^2 - \mathbf{p}_k^T \mathbf{z}\mathbf{z}^T \mathbf{p}_k = 0 \qquad \mathbf{p}_k^T \mathbf{p}_k - 1 = 0 \ . \tag{1.70}$$

The above constraints follow from the fact that an orthogonal projection of an observation, $\mathbf{z}$, onto a line, defined by $\mathbf{p}_k$ is given by $t_k = \mathbf{p}_k^T \mathbf{z}$ if $\mathbf{p}_k$ is of unit length. In a similar fashion to the formulation proposed by Anderson [2] for determining the PCA loading vectors in (1.11), (1.69) and (1.70) can be combined to produce:

$$J_k = \arg\min_{\mathbf{p}_k} \left\{ E\left\{\left(\mathbf{z} - t_k \mathbf{p}_k\right)^T \left(\mathbf{z} - t_k \mathbf{p}_k\right) - \lambda_k^{(1)} \left(t_k^2 - \mathbf{p}_k^T \mathbf{z}\mathbf{z}^T \mathbf{p}_k\right)\right\} - \lambda_k^{(2)} \left(\mathbf{p}_k^T \mathbf{p}_k - 1\right)\right\} \ . \tag{1.71}$$

Carrying out the a differentiation of J_k with respect to $\mathbf{p}_k$ yields:

$$E\left\{2t_k^2 \mathbf{p}_k - 2t_k \mathbf{z} + 2\lambda_k^{(1)} \mathbf{z}\mathbf{z}^T \mathbf{p}_k\right\} - 2\lambda_k^{(2)} \mathbf{p}_k = \mathbf{0} \ . \tag{1.72}$$

A pre-multiplication of (1.72) by $\mathbf{p}_k^T$ now reveals

$$E\left\{\underbrace{t_k^2 - \mathbf{p}_k^T \mathbf{z}\mathbf{z}^T \mathbf{p}_k}_{=0} + \lambda_k^{(1)} \underbrace{\mathbf{p}_k^T \mathbf{z}\mathbf{z}^T \mathbf{p}_k}_{=t_k^2} - \lambda_k^{(2)}\right\} = 0 \ . \tag{1.73}$$

It follows from Equation (1.73) that

$$E\left\{t_k^2\right\} = \frac{\lambda_k^{(2)}}{\lambda_k^{(1)}} \ . \tag{1.74}$$

Substituting (1.74) into Equation (1.72) gives rise to

$$\frac{\lambda_k^{(2)}}{\lambda_k^{(1)}}\mathbf{p}_k + E\left\{\lambda_k^{(1)}\mathbf{z}\mathbf{z}^T\mathbf{p}_k - \mathbf{z}\mathbf{z}^T\mathbf{p}_k\right\} - \lambda_k^{(2)}\mathbf{p}_k = \mathbf{0}\,. \tag{1.75}$$

Utilizing (1.5), the above equation can be simplified to

$$\left(\lambda_k^{(2)} - 1\right)\mathbf{S}_{ZZ}\mathbf{p}_k + \left(\frac{\lambda_k^{(2)}}{\lambda_k^{(1)}} - \lambda_k^{(2)}\right)\mathbf{p}_k = \mathbf{0}\,, \tag{1.76}$$

and, hence,

$$\left[\mathbf{S}_{ZZ} + \frac{\lambda_k^{(2)}}{\lambda_k^{(1)}}\frac{1-\lambda_k^{(1)}}{\lambda_k^{(2)}-1}\mathbf{I}\right]\mathbf{p_k} = \left[\mathbf{S}_{ZZ} - \lambda_k\mathbf{I}\right]\mathbf{p}_k = \mathbf{0} \tag{1.77}$$

with $\lambda_k = \frac{\lambda_k^{(2)}}{\lambda_k^{(1)}}\frac{1-\lambda_k^{(1)}}{\lambda_k^{(2)}-1}$. Since Equation (1.77) is identical to Equation (1.14), maximizing the variance of the score variables produces the same solution as minimizing the residual variance by orthogonally projecting the observations onto the kth weight vector. It is interesting to note that a closer analysis of Equation (1.74) yields that $E\left\{t_k^2\right\} = \frac{\lambda_k^{(2)}}{\lambda_k^{(1)}} = \lambda_k$, according to Equation (1.9), and hence, $\lambda_k^{(1)} = \frac{2}{1+\lambda_k}$ and $\lambda_k^{(2)} = 2\frac{\lambda_k}{1+\lambda_k}$, which implies that $\lambda_k^{(2)} \neq 1$ and $\frac{\frac{\lambda_k^{(2)}}{\lambda_k^{(1)}} - \lambda_k^{(2)}}{\lambda_k^{(2)}-1} = \lambda_k > 0$.

More precisely, minimizing residual variance of the projected observations and maximizing the score variance are equivalent formulations. This implies that determining a NLPCA model using a minimizing of the residual variance would produce an equivalent linear model if the nonlinear functions are simplified to be linear. This is clearly the case for principal curves and manifolds as well as the neural network approaches. In contrast, the kernel PCA approach computes a linear PCA analysis using nonlinearly transformed variables and directly addresses the variance maximization and residual minimization as per the discussion above.

Neural network approaches

It should also be noted, however, that residual variance minimization alone is a necessary but not a sufficient condition. This follows from the analysis of the ANN topology proposed by Kramer [37] in Figure 1.6. The nonlinear scores, which can extracted from the bottleneck layer, do not adhere to the fundamental principle that the first component is associated with the largest variance, the second component with the second largest variance etc. However, utilizing the sequential training of the ANN, detailed in Figure 1.7, provides an improvement, such that the first nonlinear score variables minimizes the residual variance $\mathbf{e}_1 = \mathbf{z} - \widehat{\mathbf{z}}$ and so on. However, given that the network

weights and bias terms are not subject to a length restriction as it is the case for linear PCA, this approach does also not guarantee that the first score variables possesses a maximum variance.

The same holds true for the IT network algorithm by Tan and Mavrovouniotis [68], the computed score variables do not adhere to the principal that the first one has a maximum variance. Although score variables may not be extracted that maximize a variance criterion, the computed scores can certainly be useful for feature extraction [15, 62]. Another problem of the technique by Tan and Mavrovouniotis is its application as a condition monitoring tool. Assuming the data describe a fault condition the score variables are obtained by an optimization routine to best reconstruct the fault data. It therefore follows that certain fault conditions may not be noticed. This can be illustrated using the following linear example

$$\mathbf{z}_f = \mathbf{z} + \mathbf{f} \Longrightarrow \mathbf{P}\left(\mathbf{z}_0 + \mathbf{f}\right) \ , \tag{1.78}$$

where $\mathbf{f}$ represents a step type fault superimposed on the original variable set $\mathbf{z}$ to produce the recorded fault variables $\mathbf{z}_f$. Separating the above equation produces by incorporating the statistical first order moment:

$$E\left\{\mathbf{z}_0 + \mathbf{f}_0\right\} + \mathbf{P}_0^{-T}\mathbf{P}_1^T E\left\{\mathbf{z}_1 + \mathbf{f}_1\right\} = \mathbf{P}_0^{-T}\mathbf{t} \ , \tag{1.79}$$

where the subscript $-T$ is the transpose of an inverse matrix, respectively, $\mathbf{P}^T = \left[\mathbf{P}_0^T \ \mathbf{P}_1^T\right]$, $\mathbf{z}^T = \left(\mathbf{z}_0 \ \mathbf{z}_1\right)$, $\mathbf{f}^T = \left(\mathbf{f}_0 \ \mathbf{f}_1\right)$, $\mathbf{P}_0 \in \mathbb{R}^{n\times n}$, $\mathbf{P}_1 \in \mathbb{R}^{N-n\times n}$, $\mathbf{z}_0$ and $\mathbf{f}_0 \in \mathbb{R}^N$, and $\mathbf{z}_1$ and $\mathbf{f}_1 \in \mathbb{R}^{N-n}$. Since the expectation of the original variables are zero, Equation (1.79) becomes:

$$\mathbf{f}_0 + \mathbf{P}_0^{-T}\mathbf{P}_1^T\mathbf{f}_1 = \mathbf{0} \tag{1.80}$$

which implies that if the fault vector $\mathbf{f}$ is such that $\mathbf{P}_0^{-T}\mathbf{P}_1^T\mathbf{f}_1 = -\mathbf{f}_0$ the fault condition cannot be detected using the computed score variables. However, under the assumption that the fault condition is a step type fault but the variance of $\mathbf{z}$ remains unchanged, the first order moment of the residuals would clearly be affected since

$$E\left\{\mathbf{e}\right\} = E\left\{\mathbf{z} + \mathbf{f} - \mathbf{P}\mathbf{t}\right\} = \mathbf{f} \ . \tag{1.81}$$

However, this might not hold true for an NLPCA model, where the PCA model plane, constructed from the retained loading vectors, becomes a surface. In this circumstances, it is possible to construct incipient fault conditions that remain unnoticed given that the optimization routine determines scores from the faulty observations and the IT network that minimize the mismatch between the recorded and predicted observations.

Principal curves and manifolds

By virtue of its construction, a principle curve for a two variable example can geometrically provide an NLPCA model that reduces to linear PCA if the

variable interrelationship is linear. A principal manifold, on the other hand, suffers from the same problem as neural network approaches, i.e. it is difficult to extract score variables that have the same intrinsic maximization properties as those determined by linear PCA.

Kernel PCA

Since a SVD is one mathematical tool to determine a PCA decomposition from a given data covariance or correlation matrix, the application of an SVD to the Gram matrix, to produce the score variables, will inherit the properties of linear PCA. In this respect, if the nonlinear transformation of the original variables is replaced by a linear identify mapping, kernel PCA reduces to linear PCA and therefore constitutes a true nonlinear extension to PCA.

1.5.3 Roadmap for Future Developments (Basics and Beyond)

Here, we discuss a number of issues that have only sporadically been addressed in the research literature and need, in our opinion, further attention by the research community.

Dynamic NLPCA extensions

Issues that have only been sporadically addressed are dynamic extensions of NLPCA techniques, with a notable exception being [13]. The additional computational complexity in the light of dynamic extensions mainly contributed to the lack research work being proposed thus far. However, the work in reference [13] advocates that the use of kernel PCA is a preferred technique. This confirms our analysis in the previous subsection, which raised concern about the computational demanding for training neural network based NLPCA approaches and the fact that the scores, determined by a principal manifold, do not adhere to the principal of maximum variance.

Adaptive NLPCA modeling

Adaptive modeling of nonlinear variable interrelationships is another aspects that requires a considerable research effort to develop mature, robust and efficient algorithms. This is of particular concern for process monitoring applications of systems that are time-varying and nonlinear in nature.

Parameterization of kernel functions

Although the recently proposed kernel PCA appears to be computationally efficient and maintains the properties of linear PCA, a fundamental issue that has not received considerable attention is the parameterization of the kernel

functions $\psi(\cdot)$. A general framework as to which kernel function is to be preferred for certain data pattern has not been introduced. These issues will certainly impact the performance of the kernel PCA model and need to be addressed by future research.

Extension of kernel techniques to other uniblock techniques

The extension of kernel methods to be produce nonlinear extensions other approaches that rely on the analysis of a single variable set, e.g. fisher's discriminant analysis and independent component analysis has also not received much attention in the research literature and would be an area of considerable interest for pattern, face and speech recognition as well as general feature extraction problems.

Nonlinear subspace identification

Subspace identification has been extensively studied over the past decade. This technique enables the identification of a linear state space model using input/output observations of the process. Nonlinear extensions of subspace identification have been proposed in references [76, 74, 43, 23, 41] mainly employ Hammerstein or Wiener models to represent a nonlinear steady state transformation of the process outputs. As this is restrictive, kernel PCA may be considered to determine nonlinear filters to efficiently determine this nonlinear transformation.

Variable contributions

Finally, it may be of importance to determine how an individual variable contributes to the determination of a nonlinear score variable. This issue features in process monitoring applications, where the variable contribution provides vital information for determining the root cause of abnormal process behaviour. Another area of interest is feature extraction, where a certain set of variables is responsible for a specific property observed.

1.6 Concluding Summary

This article has reviewed and critically analyzed work on nonlinear principal component analysis. The revision showed that a nonlinearity test can be applied to determine whether a conceptually and computationally more demanding NLPCA model is required. The article then showed that 3 principal directions for developing NLPCA algorithms have emerged. The first of these relate to principal curves that were initially proposed for variable sets including two variables. Extensions of principal curves are principal manifolds,

which inherit the same underlying theory. Neural network implementation represent the second research direction for implementing NLPCA. These determine nonlinear score variables either by the reduced bottleneck layer of an autoassociative neural network or by a reduced input layer whose inputs are determined by an input-training network or a predetermined principal curve. Finally, the third research direction is the recently proposed kernel PCA approach.

The analysis into (i) computational issues and (ii) their generalization of linear PCA yielded the following. Principal curves and manifolds are conceptually simple but computationally demanding for larger data and variable sets. Furthermore, whilst principal curves do produce a maximum covariance of the score variables in a similar fashion to linear PCA if only two variables are analyzed, the score obtained by a principal manifold for high-dimensional problems do not adhere to this maximization principle. NLPCA implementations based on autoassociative neural networks are cumbersome as a result of excessive computation for training the unknown network weights. Although the computationally less demanding IT networks and the incorporation of principal curves considerably reduce network complexity, neural network approaches produce nonlinear score variables that are not obtained with respect to a maximum variance criterion either. Consequently, principal manifolds and neural network approaches have been utilized in pattern, face and speech recognition, as well as feature extraction for example, they violate one fundamental principal of linear PCA, namely that of maximizing the variance of the score variables. Kernel PCA, on the other hand, apply a SVD, or a PCA analysis, on a larger set of nonlinearly transformed variables. Hence, the score variables are obtained such that the first one possesses a maximum variance, the second one the second largest variance and so on.

The paper finally outlines research areas concerning NLPCA developments and applications that require further research efforts. These include dynamic NLPCA extensions, adaptive NLPCA modelling, the parameterization of kernel functions to construct a kernel PCA model, extensions of the kernel approach to (i) other uniblock techniques such as FDI and ICA and (ii) nonlinear subspace identification and finally the evaluation of variable contributions to individual nonlinear score variables. These points have either only received sporadic attention or have not been investigated to the best of the authors knowledge.

References

1. Abdel-Qadar, I., Pashaie-Rad, S., Abudayeh, O., and Yehia, S.: PCA-based algorithm for unsupervised bridge crack detection. Advances in Engineering Software, **37** (12), 771–778 (2006)
2. Anderson, T.W.: An Introduction into Multivariate Statistical Analysis. John Wiley & Sons, New York, (1958)

3. Bakshi, B.R., Multiscale pca with application to multivariate statistical process monitoring. AIChE Journal, **44** (7), 1596–1610 (1998)
4. Banfield, J.D. and Raftery A.E.: Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. Journal of the American Statistical Association, **87** (417), 7–16 (1992)
5. Barnett, V.: Interpreting Multivariate Data. John Wiley & Sons, New York (1981)
6. Sevensen M., Bishop, C.M., and Williams C.K.I.: GTM: The generative topographic mapping. Neural Computation, **10**, 215–234 (1998)
7. Chang, K. and Ghosh, J.: Principal curve classifier - a nonlinear approach to pattern classification. In: IEEE International Joint Conference on Neural Networks, 4-9 May 1998, 695–700, Anchorage, Alaska (1998)
8. Chang, K. and Ghosh, J.: A unified model for probabilistic principal surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, **23** (1), 22–41 (2001)
9. Chen, D., Zhang, J., Tang, S., and Wang J.: Freeway traffic stream modelling based on principal curves and its analysis. IEEE Transactions on Intelligent Transportation Systems, **5** (4), 246–258 (2004)
10. Chen, P. and Suter, D.: An analysis of linear subspace approaches for computer vision and pattern recognition. International Journal of Computer Vision, **68** (1), 83–106 (2006)
11. Chennubhotla, C. and Jepson, A.: Sparse pca extracting multi-scale structure from data. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 1, 641–647 (2001)
12. Cho, H.W.: Nonlinear feature extraction and classification of multivariate process data in kernel feature space. Expert Systems with Applications, **32** (2), 534–542 (2007)
13. Choi, S.W. and Lee, I.-B.: Nonlinear dynamic process monitoring based on dynamic kernel pca. Chemical Engineering Science, **59** (24), 5897–5908 (2004)
14. Delicado, P.: Another look at principal curves and surfaces. Journal of Multivariate Analysis, **77** (1), 84–116 (2001)
15. Denoeux, T. and Masson, M.-H.: Principal component analysis of fuzzy data using autoassociative neural networks. IEEE Transactions on Fuzzy Systems, **12** (3), 336–349 (2004)
16. Dong, D. and McAvoy, T.J.: Nonlinear principal component analysis–based on principal curves and neural networks. Computers & Chemical Engineering, **20** (1), 65–78 (1996)
17. Du, Q. and Chang, C.: Linear mixture analysis-based compression for hyperspectal image analysis. IEEE Transactions on Geoscience and Remote Sensing, **42** (4), 875–891 (2004)
18. Duchamp, T. and Stuetzle, W.: Extremal properties of principal curves in the plane. Annals of Statistics, **24** (4), 1511–1520 (1996)
19. Duchamp, T. and Stuetzle, W.: Geometric Properties of Principal Curves in the Plane. In: Robust statistics, data analysis, and computer intensive methods: in honor of Peter Huber's 60th birthday, (Lecture Notes in Statistics), vol. 109, 135–152. Springer, New York (1996)
20. Esbensen, K. and Geladi, P.: Strategy of multivariate image analysis (MIA). Chemometrics & Intelligent Laboratory Systems, **7** (1-2), 67–86 (1989)

21. Fisher, R. and MacKenzie, W.: Studies in crop variation, ii, the manurial response of different potato varieties. Journal of Agricultural Science, **13**, 411–444 (1923)
22. Golub, G.H. and van Loan, C.F.: Matrix Computation. John Hopkins, Baltimore, (1996)
23. Gomez, J.C. and Baeyens, E.: Subspace-based identification algorithms for hammerstein and wiener models. European Journal of Control, **11** (2), 127–136 (2005)
24. Hastie, T.: Principal curves and surfaces. Technical report no. 11, Department of Statistics, Stanford University (1984)
25. Hastie, T. and Stuetzle, W.: Principal curves. Journal of the American Statistical Association **84** (406), 502–516 (1989)
26. Hermann, T, Meinicke, P., and Ritter, H.: Principal curve sonification. In: Proceedings of International Conference on Auditory Display, 2-5 April 2000, Atlanta, Georgia, 81–86 (2000)
27. Hertz, J., Krogh, A., and Palmer, R.G.: Introduction to the Theory of Neural Computing. Addison-Wesley, Redwood City, CA (1991)
28. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, **24** 417–441 (1933)
29. Jackson, J.E.: Principal components and factor analysis: Part III: What is factor analysis. Journal of Quality Technology, **13** (2), 125–130 (1981)
30. Jackson, J.E.: A Users Guide to Principal Components. Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York (1991)
31. Jia, F., Martin, E.B., and Morris, A.J.: Non-linear principal component analysis for process fault detection. Computers & Chemical Engineering, **22** (Supplement), S851–S854 (1998)
32. Joliffe, I.T.: Principal Component Analysis. Springer, New York, (1986)
33. Kégl, B.: Principal Curves: Learning, Design and Applications. PhD thesis, Department of Computer Science, Concordia University, Montréal, Québec, Canada, 2000
34. Kégl, B., Krzyzak, A., Linder, T., and Zeger, K.: A polygonal line algorithm for constructing principal curves. In: Neural Information Processing (NIPS '98), Denver, CO, 1-3 December 1998, 501–507. MIT Press (1998)
35. Kégl, B., Krzyzak, A., Linder, T., and Zeger, K.: Learning and design of principal curves. IEEE Transactions on Pattern Analysis and Machine Intelligence, **22** (3), 281–297 (2000)
36. Kim, K.I., Jung, K., and Kim, H. J.: Face recognition using kernel principal component analysis. IEEE Signal Processing Letters, **9** (2), 40–42 (2002)
37. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal, **37** (3), 233–243 (1991)
38. Kruger, U., Antory, D., Hahn, J., Irwin, G.W., and McCullough, G.: Introduction of a nonlinearity measure for principal component models. Computers & Chemical Engineering, **29** (11-12), 2355–2362 (2005)
39. Krzanowski, W. J.: Cross-validatory choice in principal component analysis: Some sampling results. Journal of Statistical Computation and Simulation, **18**, 299–314 (1983)
40. Kwok, J.T.Y. and Tsang, I.W.H.: The pre-image problem in kernel methods. IEEE Transactions on Neural Networks, **15** (6), 1517–1525 (2004)

41. Lacy, S.L. and Bernstein, D. S.: Subspace identification for non-linear systems with measured input non-linearities. International Journal of Control, **78** (12), 906–921 (2005)
42. Leeuw J. d.: Nonlinear principal component analysis. In: Caussinus, H., Ettinger, P., and Tomassone, R. (eds) Proceedings in Computational Statistics (COMPSTAT 1982) October 30 – September 3, Toulouse, France 1982. Physica-Verlag, Wien (1982)
43. Lovera, M., Gustafsson, T., and Verhaegen, M.: Recursive subspace identification of linear and nonlinear wiener type models. Automatica, **36** (11), 1639–1650 (2000)
44. Malinowski, E.R.: Factor Analysis in Chemistry. John Wiley & Sons, New York (2002)
45. Mardia, K.V., Kent, J.T., and Bibby, J.M.: Multivariate Analysis. Probability and Mathematical Statistics. Academic Press, London, (1979)
46. Morales, M.: Geometric Data Fitting. PhD thesis, University of Washington (1998)
47. Nara, Y., Jianming Y., and Suematsu, Y.: Face recognition using improved principal component analysis. In: Proceedings of 2003 International Symposium on Micromechatronics and Human Science (IEEE Cat. No.03TH8717), 77–82 (2003)
48. Nomikos, P. and MacGregor, J.F.: Monitoring of batch processes using multiway principal component analysis. AIChE Journal, **40** (8), 1361–1375 (1994)
49. Paluš, M. and Dvořák, I.: Singular-value decomposition in attractor reconstruction: Pitfals and precautions. Physica D: Nonlinear Phenomena, **5** (1-2), 221–234 (1992)
50. Parsopoulos, K.E. and Vrahatis, M. N.: Recent approaches to global optimization problems through particle swarm optimization. Natural Computing, **1** (2-3), 235–306 (2002)
51. Pearson, C.: On lines and planes of closest fit to systems of points in space. Phil. Mag., Series B., **2** (11), 559–572 (1901)
52. Qin, S.J., Valle, S., and Piovoso, M. J.: On unifying multi-block analysis with application to decentralized process monitoring. Journal of Chemometrics, **10**, 715–742 (2001)
53. Reinhard, K., and Niranjan, M.: Subspace models for speech transitions using principal curves. Proceedings of Institute of Acoustics, **20** (6), 53–60 (1998)
54. Reinhard, K., and Niranjan, M.: Parametric subspace modeling of speech transitions. Speech Communication, **27** (1), 19–42 (1999)
55. Sandilya, S. and Kulkarni, S.R.: Principal curves with bounded turn. In: Proceedings of the IEEE International Symposium on Information Theory, Sorento, 25-30 June 2000, Sorento, Italy (2000)
56. Schölkopf, B. and Smola, A. J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, Cambridge, MA (2002)
57. Schölkopf, B. and Smola, A. J., and Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, **10** (5), 1299–1319 (1998)
58. Shanmugam, R. and Johnson, C.: At a crossroad of data envelopment and principal component analyses. Omega, **35** (4), 351–364 (2007)
59. Shawe-Taylor, J. and Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, West Nyack, NY (2004)

60. Silverman, B. W.: Some aspects of spline smoothing. Journal of the Royal Statistical Society, Series B, **47** (1), 1–52 (1985)
61. Smola, A.J., Williamson, R.C., Mika, S., and Schölkopf, B.: Regularized principal manifolds. In: Fischer, P., and Simon, H.U. (eds.) Computational Learning Theory (EuroCOLT '99), Lecture Notes in Artificial Intelligence, vol. 1572, 214–229, Springer, Heidelberg (1999)
62. Socas-Navarro, H.: Feature extraction techniques for the analysis of spectral polarization profiles. Astrophysical Journal, **620**, 517–522 (2005)
63. Srinivas, M. and Patnaik, L.M.: Genetic algorithms: A survey. Computer, **27** (6), 17–26 (1994)
64. Stoica, P., Eykhoff, P., Janssen, P., and Söderström, T.: Model structure selection by cross-validation. International Journal of Control, **43** (6), 1841–1878 (1986)
65. Stone, M.: Cross-validatory choice and assessment of statistical prediction (with discussion). Journal of the Royal Statistical Society (Series B), **36**, 111–133 (1974)
66. Stoyanova, R. and Brown, T. R.: Nmr spectral quantitation by principal component analysis. NMR in Biomedicine, **14** (4), 271–277 (2001)
67. Sylvester, J.J.: On the reduction of a bilinear quantic of the nth order to the form of a sum of n products by a double orthogonal substitution. Messenger of Mathematics, **19**, 42–46 (1889)
68. Tan, S. and Mavrovouniotis, M.L.: Reducing data dimensionality through optimizing neural network inputs. AIChE Journal, **41** (6), 1471–1480 (1995)
69. Tibshirani, R.: Principal curves revisited. Statistics and Computation, **2** (4), 183–190 (1992)
70. Trafalis, T.B., Richman, M.B., White, A., and Santosa, B.: Data mining techniques for improved wsr-88d rainfall estimation. Computers & Industrial Engineering, **43** (4), 775–786 (2002)
71. Huffel, S. van, and Vandewalle, J.: The Total Least Squares Problem: Computational Aspects and Analysis. SIAM, Philadelphia (1991)
72. Vaswani, N. and Chellappa, R.: Principal components null space analysis for image and video classification. IEEE Transactions on Image Processing, **15** (7), 1816–1830 (2006)
73. Vlassis, N., Verbeek, J.J., and Kr´ose, B.: K-segments algorithm for finding principal curves. Technical Report IAS-UVA-00-11, Institute of Computer Science, University of Amsterdam (2000)
74. Verhaegen, M. and Westwick, D.: Identifying mimo hammerstein systems in the context of subspace model identification methods. International Journal of Control, **63** (2), 331–350 (1996)
75. Wax, M. and Kailath, T.: Detection of signals by information theoretic criteria. IEEE Transactions on Acoustics, Speech & Signal Processing, **33** (2), 387–392 (1985)
76. Westwick, D. and Verhaegen, M.: Identifying mimo wiener systems using subspace model identification methods. Signal Processing, **52** (2), 235–258 (1996)
77. Wise, B.M. and Ricker, N.L.: Identification of finite impulse respnose models by principal components regression: Frequency response properties. Process Control & Quality, **4**, 77–86 (1992)
78. Wold, H.: Estimation of principal components and related models by iterative least squares. In: Krishnaiah, P.R. (ed.) Multivariate Analysis, 391–420. Academic Press, N.Y. (1966)

79. Wold, S.: Cross validatory estimation of the number of principal components in factor and principal component models. Technometrics, **20** (4), 397–406 (1978)
80. Wold, S., Esbensen, K., and Geladi, P.: Principal component analysis. Chemometrics and Intelligent Laboratory Systems, **2**, 37–52 (1987)
81. Yoo, C.K. and Lee, I.: Nonlinear multivariate filtering and bioprocess monitoring for supervising nonlinear biological processes. Process Biochemistry, **41** (8), 1854–1863 (2006)
82. Zeng, Z. and Zou, X.: Application of principal component analysis to champ radio occultation data for quality control and a diagnostic study. Monthly Weather Review, **134** (11), 3263–3282 (2006)
83. Zhang, J. and Chen, D.: Constraint k-segment principal curves. In: Huang, De-Sh., Li, K. and Irwin, G.W. (eds.) Intelligent Computing, Lecture Notes in Computer Sciences, vol. 4113, 345–350. Springer, Berlin Heidelberg New York (2006)
84. Zhang, J., Chen, D., and Kruger, U.: Constrained k-segments principal curves and its applications in intelligent transportation systems. Technical report, Department of Computer Science and Engineering, Fudan University, Shanghai, P. R. China (2007)
85. Zhang, J. and Wang, J.: An overview of principal curves (in chinese). Chinese Journal of Computers, **26** (2), 137–148 (2003)