8

Auto-Associative Models, Nonlinear Principal Component Analysis, Manifolds and Projection Pursuit

Stéphane Girard[1] and Serge Iovleff[2]

[1] INRIA Rhône-Alpes, projet Mistis, Inovallée, 655, av. de l'Europe, Montbonnot, 38334 Saint-Ismier cedex, France,
Stephane.Girard@inrialpes.fr
[2] Laboratoire Paul Painlevé, 59655 Villeneuve d'Ascq Cedex, France,
serge.iovleff@univ-lille1.fr

Summary. Auto-associative models have been introduced as a new tool for building nonlinear Principal component analysis (PCA) methods. Such models rely on successive approximations of a dataset by manifolds of increasing dimensions. In this chapter, we propose a precise theoretical comparison between PCA and auto-associative models. We also highlight the links between auto-associative models, projection pursuit algorithms, and some neural network approaches. Numerical results are presented on simulated and real datasets.

8.1 Introduction

Principal component analysis (PCA) is a well-known method for extracting linear structures from high-dimensional datasets. It computes the subspace best approaching the dataset from the Euclidean point of view. This method benefits from efficient implementations based either on solving an eigenvalue problem or on iterative algorithms. We refer to [27] for details. In a similar fashion, multi-dimensional scaling [3, 35, 44] addresses the problem of finding the linear subspace best preserving the pairwise distances. More recently, new algorithms have been proposed to compute low dimensional embeddings of high dimensional data. For instance, Isomap [46], LLE (Locally linear embedding) [42] and CDA (Curvilinear distance analysis) [9] aim at reproducing in the projection space the structure of the initial local neighborhood. These methods are mainly dedicated to visualization purposes. They cannot produce an analytic form of the transformation function, making it difficult to map new points into the dimensionality-reduced space. Besides, since they rely on local properties of pairwise distances, these methods are sensitive to noise and outliers. We refer to [38] for a comparison between Isomap and CDA and to [48] for a comparison between some features of LLE and Isomap.

Finding nonlinear structures is a challenging problem. An important family of methods focuses on self-consistent structures. The self-consistency concept is precisely defined in [45]. Geometrically speaking, it means that each point of the structure is the mean of all points that project orthogonally onto it. For instance, it can be shown that the K-Means algorithm [23] converges to a set of k self-consistent points. Principal curves and surfaces [8, 24, 37, 47] are examples of one-dimensional and two-dimensional self-consistent structures. Their practical computation requires to solve a nonlinear optimization problem. The solution is usually non robust and suffers from a high estimation bias. In [31], a polygonal algorithm is proposed to reduce this bias. Higher dimensional self-consistent structures are often referred to as self-consistent manifolds even though their existence is not guaranteed for arbitrary datasets. An estimation algorithm based on a grid approximation is proposed in [19]. The fitting criterion involves two smoothness penalty terms describing the elastic properties of the manifold.

In this paper, auto-associative models are proposed as candidates to the generalization of PCA. We show in paragraph 8.2.1 that these models are dedicated to the approximation of the dataset by a manifold. Here, the word "manifold" refers to the topology properties of the structure [39]. The approximating manifold is built by a projection pursuit algorithm presented in paragraph 8.2.2. At each step of the algorithm, the dimension of the manifold is incremented. Some theoretical properties are provided in paragraph 8.2.3. In particular, we can show that, at each step of the algorithm, the mean residuals norm is not increased. Moreover, it is also established that the algorithm converges in a finite number of steps. Section 8.3 is devoted to the presentation of some particular auto-associative models. They are compared to the classical PCA and some neural networks models. Implementation aspects are discussed in Section 8.4. We show that, in numerous cases, no optimization procedure is required. Some illustrations on simulated and real data are presented in Section 8.5.

8.2 Auto-Associative Models

In this chapter, for each unit vector $a \in \mathbb{R}^p$, we denote by $P_a(.) = \langle a, . \rangle$ the linear projection from $\mathbb{R}^p$ to $\mathbb{R}$. Besides, for all set E, the identity function $E \to E$ is denoted by Id_E.

8.2.1 Approximation by Manifolds

A function F^d: $\mathbb{R}^p \to \mathbb{R}^p$ is a d-dimensional auto-associative function if there exist d unit orthogonal vectors a^k, called principal directions, and d continuously differentiable functions s^k: $\mathbb{R} \to \mathbb{R}^p$, called regression functions, such that

$$P_{a^j} \circ s^k = \delta_{j,k} Id_{\mathbb{R}} \quad \text{for all} \quad 1 \leq j \leq k \leq d \,, \tag{8.1}$$

where $\delta_{j,k}$ is the Kronecker symbol and

$$F^d = \left(Id_{\mathbb{R}^p} - s^d \circ P_{a^d}\right) \circ \ldots \circ \left(Id_{\mathbb{R}^p} - s^1 \circ P_{a^1}\right) = \coprod_{k=d}^{1} \left(Id_{\mathbb{R}^p} - s^k \circ P_{a^k}\right) \ . \tag{8.2}$$

The main feature of auto-associative functions is mainly a consequence of (8.1):

Theorem 1. *The equation $F^d(x) = 0$, $x \in \mathbb{R}^p$ defines a differentiable d-dimensional manifold of $\mathbb{R}^p$.*

We refer to [16] for a proof. Thus, the equation $F^d(x) = 0$ defines a space in which every point has a neighborhood which resembles the Euclidean space $\mathbb{R}^d$, but in which the global structure may be more complicated. As an example, on a 1-dimensional manifold, every point has a neighborhood that resembles a line. In a 2-manifold, every point has a neighborhood that looks like a plane. Examples include the sphere or the surface of a torus.

Now, let X be a square integrable random vector of $\mathbb{R}^p$. Assume, without loss of generality, that X is centered and introduce $\sigma^2(X) \stackrel{\text{def}}{=} \mathbb{E}[\|X\|^2]$. For all auto-associative function F^d, let us consider $\varepsilon = F^d(X)$. Note that, from the results of Subsection 8.2.3 below, ε is necessarily a centered random vector. In this context, $\sigma^2(\varepsilon)$ is called the residual variance. Geometrically speaking, the realizations of the random vector X are approximated by the manifold $F^d(x) = 0$, $x \in \mathbb{R}^p$ and $\sigma^2(\varepsilon)$ represents the variance of X "outside" the manifold.

Of course, such random vector X always satisfies a 0-dimensional auto-associative model with $F^0 = Id_{\mathbb{R}^p}$ and $\sigma^2(\varepsilon) = \sigma^2(X)$. Similarly, X always satisfies a p-dimensional auto-associative model with $F^p = 0$ and $\sigma^2(\varepsilon) = 0$. In practice, it is important to find a balance between these two extreme cases by constructing a d-dimensional model with $d \ll p$ and $\sigma^2(\varepsilon) \ll \sigma^2(X)$. For instance, in the case where the covariance matrix Σ of X is of rank d, then X is located on a d-dimensional linear subspace defined by the equation $F^d_{PCA}(x) = 0$ with

$$F^d_{PCA}(x) = x - \sum_{k=1}^{d} P_{a^k}(x) a^k \ , \tag{8.3}$$

and where a^k, $k = 1, \ldots, d$ are the eigenvectors of Σ associated to the positive eigenvalues. A little algebra shows that (8.3) can be rewritten as $F^d(x) = 0$, where F^d is a d-dimensional auto-associative function with linear regression functions $a^k(t) = ta^k$ for $k = 1, \ldots, d$. Moreover, we have $\sigma^2(\varepsilon) = 0$. Since (8.3) is the model produced by a PCA, it straightforwardly follows that PCA is a special (linear) case of auto-associative models. In the next section, we propose an algorithm to build auto-associative models with non necessarily linear regression functions, small dimension and small residual variance. Such

models could also be called "semi-linear" or "semi-parametric" since they include a linear/parametric part through the use of linear projection operators and a non-linear/non-parametric part through the regression functions.

8.2.2 A Projection Pursuit Algorithm

Let us recall that, given an unit vector $a \in \mathbb{R}^p$, an index I: $\mathbb{R} \to \mathbb{R}$ is a functional measuring the interest of the projection $P_a(X)$ with a non negative real number. The meaning of the word "interest" depends on the considered data analysis problem. For instance, a possible choice of I is the projected variance $I \circ P_a(.) = \mathrm{Var}[P_a(.)]$. Some other examples are presented in Section 8.4.2. Thus, the maximization of $I \circ P_a(X)$ with respect to a yields the most interesting direction for this given criteria. An algorithm performing such an optimization is called a projection pursuit algorithm. We refer to [26] and [28] for a review on this topic.

Let $d \in \{0, \ldots, p\}$, and consider the following algorithm which consists in applying iteratively the following steps: [A] computation of the Axes, [P] Projection, [R] Regression and [U] Update:

Algorithm 1 *Define* $R^0 = X$.
For $k = 1, \ldots, d$*:*

[A] *Determine* $a^k = \arg\max_{x \in \mathbb{R}^p} I \circ P_x(R^{k-1})$ *s.t.* $\|x\| = 1$, $P_{a^j}(x) = 0$, $1 \leq j < k$.
[P] *Compute* $Y^k = P_{a^k}(R^{k-1})$.
[R] *Estimate* $s^k(t) = \mathbb{E}[R^{k-1} | Y^k = t]$,
[U] *Compute* $R^k = R^{k-1} - s^k(Y^k)$.

The random variables Y^k are called principal variables and the random vectors R^k residuals. Step [A] consists in computing an axis orthogonal to the previous ones and maximizing a given index I. Step [P] consists in projecting the residuals on this axis to determine the principal variables, and step [R] is devoted to the estimation of the regression function of the principal variables best approximating the residuals. Step [U] simply consists in updating the residuals. Thus, Algorithm 1 can be seen as a projection pursuit regression algorithm [14, 32] since it combines a projection pursuit step [A] and a regression step [R]. The main problem of such approaches is to define an efficient way to iterate from k to $k+1$. Here, the key property is that the residuals R^k are orthogonal to the axis a^k since

$$\begin{aligned} P_{a^k}(R^k) &= P_{a^k}(R^{k-1}) - P_{a^k} \circ s^k(Y^k) \\ &= P_{a^k}(R^{k-1}) - \mathbb{E}[P_{a_k}(R^{k-1}) | Y^k] \\ &= Y^k - \mathbb{E}[Y^k | Y^k] \\ &= 0\,. \end{aligned} \tag{8.4}$$

Thus, it is natural to iterate the model construction in the subspace orthogonal to a^k, see the orthogonality constraint in step [A]. The theoretical results provided in the next paragraph are mainly consequences of this property.

8.2.3 Theoretical Results

Basing on (8.4), it is easily shown by induction that both the residuals and the regression functions computed at the iteration k are almost surely (a.s.) orthogonal to the axes computed before. More precisely, one has

$$\left\langle a^j, R^k \right\rangle = 0 \ , \ a.s. \ \text{ for all } \ 1 \leq j \leq k \leq d \ , \tag{8.5}$$

$$\left\langle a^j, s^k(Y^k) \right\rangle = 0, \ a.s. \ \text{ for all } \ 1 \leq j < k \leq d \ . \tag{8.6}$$

Besides, the residuals, principal variables and regression functions are centered:

$$\mathbb{E}[R^k] = \mathbb{E}[Y^k] = \mathbb{E}[s^k(Y^k)] = 0 \ ,$$

for all $1 \leq k \leq d$. Our main result is the following:

Theorem 2. *Algorithm 1 builds a d-dimensional auto-associative model with principal directions* $\{a^1, \ldots, a^d\}$*, regression functions* $\{s^1, \ldots, s^d\}$ *and residual* $\varepsilon = R^d$*. Moreover, one has the expansion*

$$X = \sum_{k=1}^{d} s^k(Y^k) + R^d \ , \tag{8.7}$$

where the principal variables Y^k *and* Y^{k+1} *are centered and non-correlated for* $k = 1, \ldots, d-1$*.*

The proof is a direct consequence of the orthogonality properties (8.5) and (8.6). Let us highlight that, for $d = p$, expansion (8.7) yields an exact expansion of the random vector X as:

$$X = \sum_{k=1}^{p} s^k(Y^k) \ ,$$

since $R^p = 0$ (a.s.) in view of (8.5). Finally, note that the approximation properties of the conditional expectation entails that the sequence of the residual norms is almost surely non increasing. As a consequence, the following corollary will prove useful to select the model dimension similarly to the PCA case.

Corollary 1. *Let* Q_d *be the information ratio represented by the d-dimensional auto-associative model:*

$$Q_d = 1 - \sigma^2(R^d) \, / \sigma^2(X) \ .$$

Then, $Q_0 = 0$*,* $Q_p = 1$ *and the sequence* (Q_d) *is non decreasing.*

Note that all these properties are quite general, since they do not depend either on the index I, nor on the estimation method for the conditional expectation. In the next section, we show how, in particular cases, additional properties can be obtained.

8.3 Examples

We first focus on the auto-associative models which can be obtained using linear estimators of the regression functions. The existing links with PCA are highlighted. Second, we introduce the intermediate class of additive auto-associative models and compare it to some neural network approaches.

8.3.1 Linear Auto-Associative Models and PCA

Here, we limit ourselves to linear estimators of the conditional expectation in step [R]. At iteration k, we thus assume

$$s^k(t) = tb^k,\ t \in \mathbb{R}\ , b^k \in \mathbb{R}^p\ .$$

Standard optimization arguments (see [18], Proposition 2) shows that, necessarily, the regression function obtained at step [R] is located on the axis

$$b^k = \Sigma_{k-1}a^k/({}^t a^k \Sigma_{k-1} a^k)\ , \tag{8.8}$$

with Σ_{k-1} the covariance matrix of R^{k-1}:

$$\Sigma_{k-1} = \mathbb{E}[R^{k-1}\,{}^t R^{k-1}]\ , \tag{8.9}$$

and where, for all matrix M, the transposed matrix is denoted by ${}^t M$. As a consequence of Theorem 2, we have the following linear expansion:

$$X = \sum_{k=1}^{d} \frac{Y^k \Sigma_{k-1} a^k}{{}^t a^k \Sigma_{k-1} a^k} + R^d\ .$$

As an interesting additional property of these so-called linear auto-associative models, we have $\mathbb{E}[Y_j Y_k] = 0$ for all $1 \leq j < k \leq d$. This property is established in [18], Proposition 2. Therefore, the limitation to a family of linear functions in step [R] allows to recover an important property of PCA models: the non-correlation of the principal variables. It is now shown that Algorithm 1 can also compute a PCA model for a well suited choice of the index.

Proposition 1. *If the index in step [A] is the projected variance, i.e.*

$$I \circ P_x(R^{k-1}) = \mathrm{Var}[P_x(R^{k-1})]\ ,$$

and step [R] is given by (8.8) then Algorithm 1 computes the PCA model of X.

Indeed, the solution a^k of step [A] is the eigenvector associated to the maximum eigenvalue of Σ_{k-1}. From (8.8) it follows that $b^k = a^k$. Replacing in (8.2), we obtain, for orthogonality reasons, $F^d = F^d_{PCA}$.

8.3.2 Additive Auto-Associative Models and Neural Networks

A d-dimensional auto-associative function is called additive if (8.2) can be rewritten as

$$F^d = Id_{\mathbb{R}^d} - \sum_{k=1}^{d} s^k \circ P_{a^k} \ . \tag{8.10}$$

In [17], the following characterization of additive auto-associative functions is provided. A d-dimensional auto-associative function is additive if and only if

$$P_{a^j} \circ s^k = \delta_{j,k} Id_{\mathbb{R}} \text{ for all } (j,k) \in \{1,\ldots,d\}^2 \ .$$

As a consequence, we have:

Theorem 3. *In the linear subspace spanned by $\{a^1,\ldots,a^d\}$, every d-dimensional additive auto-associative model reduces to the PCA model.*

A similar result can be established for the nonlinear PCA based on a neural network and introduced in [29]. The proposed model is obtained by introducing a nonlinear function $g : \mathbb{R} \to \mathbb{R}$, called activation function, in the PCA model (8.3) to obtain

$$F^d_{KJ}(x) = x - \sum_{k=1}^{d} g \circ P_{a^k}(x) a^k \ . \tag{8.11}$$

Note that (8.11) is an additive auto-associative model as defined in (8.10) if and only if $g = Id_{\mathbb{R}}$, *i.e.* if and only if it reduces to the PCA model in the linear subspace spanned by $\{a^1,\ldots,a^d\}$. Moreover, in all cases, we have

$$\{F^d_{KJ}(x) = 0,\ x \in \mathbb{R}^p\} \subset \{F^d_{PCA}(x) = 0,\ x \in \mathbb{R}^p\} \ ,$$

which means that this model is included in the PCA one. More generally, the auto-associative Perceptron with one hidden layer [7] is based on multidimensional activation functions $\sigma^k : \mathbb{R} \to \mathbb{R}^p$:

$$F^d_{AAP}(x) = x - \sum_{k=1}^{d} \sigma^k \circ P_{a^k}(x) \ . \tag{8.12}$$

Unfortunately, it can be shown [10] that a single hidden layer is not sufficient. Linear activation functions (leading to a PCA) already yield the best approximation of the data. In other words, the nonlinearity introduced in (8.12) has no significant effect on the final approximation of the dataset. Besides, determining a^k, $k = 1,\ldots,d$ is a highly nonlinear problem with numerous local minima, and thus very dependent on the initialization.

8.4 Implementation Aspects

In this section, we focus on the implementation aspects associated to Algorithm 1. Starting from a n-sample $\{X_1, \ldots, X_n\}$, two problems are addressed. In Subsection 8.4.1, we propose some simple methods to estimate the regression functions s^k appearing in step [R]. In Subsection 8.4.2, the choice of the index in step [A] is discussed. In particular, we propose a contiguity index whose maximization is explicit.

8.4.1 Estimation of the Regression Functions

Linear auto-associative models

To estimate the regression functions, the simplest solution is to use a linear approach leading to a linear auto-associative model. In this case, the regression axis is explicit, see (8.8), and it suffices to replace Σ_{k-1} defined in (8.9) by its empirical counterpart

$$V_{k-1} = \frac{1}{n}\sum_{i=1}^{n} R_i^{k-1\,t} R_i^{k-1}\ , \tag{8.13}$$

where R_i^{k-1} is the residual associated to X_i at iteration $k-1$.

Nonlinear auto-associative models

Let us now focus on nonlinear estimators of the conditional expectation $s^k(t) = \mathbb{E}[R^{k-1}|Y^k = t]$, $t \in \mathbb{R}$. Let us highlight that s^k is a univariate function and thus its estimation does not suffer from the curse of dimensionality [1]. This important property is a consequence of the "bottleneck" trick used in (8.2) and, more generally, in neural networks approaches. The key point is that, even though $s^k \circ P_{a^k}$ is a p- variate function, its construction only requires the nonparametric estimation of a univariate function thanks to the projection operator.

For the sake of simplicity, we propose to work in the orthogonal basis B^k of $\mathbb{R}^p$ obtained by completing $\{a^1, \ldots, a^k\}$. Let us denote by R_j^{k-1} the j-th coordinate of R^{k-1} in B^k. In view of (8.5), $R_j^{k-1} = 0$ for $j = 1, \ldots, k-1$. Besides, from step [P], $R_k^{k-1} = Y^k$. Thus, the estimation of $s^k(t)$ reduces to the estimation of $p-k$ functions

$$s_j^k(t) = \mathbb{E}[R_j^{k-1}|Y^k = t], \quad j = k+1, \ldots, p\ .$$

This standard problem [22, 12] can be tackled either by kernel [2] or projection [20] estimates.

Kernel estimates

Each coordinate $j \in \{k+1, \ldots, p\}$ of the estimator can be written in the basis B^j as:

$$\hat{s}_j^k(t) = \sum_{i=1}^{n} R_{j,i}^{k-1} K\left(\frac{t - Y_i^k}{h}\right) \Bigg/ \sum_{i=1}^{n} K\left(\frac{t - Y_i^k}{h}\right) \ , \tag{8.14}$$

where $R_{j,i}^{k-1}$ represents the j-th coordinate of the residual associated to the observation X_i at the $(k-1)$-th iteration in the basis B^k, Y_i^k is the value of the k-th principal variable for the observation X_i and K is a Parzen-Rosenblatt kernel, that is to say a bounded real function, integrating to one and such that $tK(t) \to 0$ as $|t| \to \infty$. For instance, one may use a a standard Gaussian density. The parameter h is a positive number called window in this context. In fact, $\hat{s}_j^k(t)$ can be seen as a weighted mean of the residuals $R_{j,i}^{k-1}$ which are close to t:

$$\hat{s}_j^k(t) = \sum_{i=1}^{n} R_{j,i}^{k-1} w_i^k(t) \ ,$$

where the weights are defined by

$$w_i^k(t) = K\left(\frac{t - Y_i^k}{h}\right) \Bigg/ \sum_{i=1}^{n} K\left(\frac{t - Y_i^k}{h}\right) \ ,$$

and are summing to one:

$$\sum_{i=1}^{n} w_i^k(t) = 1 \ .$$

The amplitude of the smoothing is tuned by h. In the case of a kernel with bounded support, for instance if $\mathrm{supp}(K) = [-1, 1]$, the smoothing is performed on an interval of length $2h$. For an automatic choice of the smoothing parameter h, we refer to [25], Chapter 6.

Projection estimates

Each coordinate $j \in \{k+1, \ldots, p\}$ of the estimator is expanded on a basis of L real functions $\{b_\ell(t),\ \ell = 1, \ldots, L\}$ as:

$$\tilde{s}_j^k(t) = \sum_{\ell=1}^{L} \tilde{\alpha}_{j,\ell}^k b_\ell(t) \ .$$

The coefficients $\tilde{\alpha}_{j,\ell}^k$ appearing in the linear combination of basis functions are determined such that $\tilde{s}_j^k(Y_i^k) \simeq R_{j,i}^{k-1}$ for $i = 1, \ldots, n$. More precisely,

$$\tilde{\alpha}_{j,\cdot}^k = \arg\min_{\alpha_{j,\cdot}^k} \sum_{i=1}^{n} \left(\sum_{\ell=1}^{L} \alpha_{j,\ell}^k b_\ell(Y_i^k) - R_{j,i}^{k-1} \right)^2 \ ,$$

and it is well-known that this least-square problem benefits from an explicit solution which can be matricially written as

$$\tilde{\alpha}^k_{j,\cdot} = ({}^tB^kB^k)^{-1}\,{}^tB^kR^{k-1}_{j,\cdot}\ , \tag{8.15}$$

where B^k is the $n \times L$ matrix with coefficients $B^k_{i,\ell} = b_\ell(Y^k_i)$, $i = 1, \dots n$, $\ell = 1, \dots, L$. Note that this matrix does not depend on the coordinate j. Thus, the matrix inversion in (8.15) is performed only once at each iteration k. Besides, the size of this matrix is $L \times L$ and thus does not depend either on the dimension of the space p, nor on the sample size n. As an example, one can use a basis of cubic splines [11]. In this case, the parameter L is directly linked to N the number of knots: $L = N + 4$. Remark that, in this case, condition $N + 4 \leq n$ is required so that the matrix is ${}^tB^kB^k$ is regular.

8.4.2 Computation of Principal Directions

The choice of the index I is the key point of any projection pursuit problem where it is needed to find "interesting" directions. We refer to [26] and [28] for a review on this topic. Let us recall that the meaning of the word "interesting" depends on the considered data analysis problem. As mentioned in Subsection 8.2.2, the most popular index is the projected variance

$$I_{PCA} \circ P_x\left(R^{k-1}\right) = \frac{1}{n}\sum_{i=1}^{n} P_x^2(R^{k-1}_i) \tag{8.16}$$

used in PCA. Remarking that this index can be rewritten as

$$I_{PCA} \circ P_x\left(R^{k-1}\right) = \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j\neq i} P_x^2(R^{k-1}_i - R^{k-1}_j)\ ,$$

it appears that the "optimal" axis maximizes the mean distance between the projected points. An attractive feature of the index (8.16) is that its maximization benefits from an explicit solution in terms of the eigenvectors of the empirical covariance matrix V_{k-1} defined in (8.13). Friedman *et al* [15, 13], and more recently Hall [21], proposed an index to find clusters or use deviation from the normality measures to reveal more complex structures of the scatter-plot. An alternative approach can be found in [4] where a particular metric is introduced in PCA so as to detect clusters. We can also mention indices dedicated to outliers detection [40]. Similar problems occur in the neural networks context where the focus is on the construction of nonlinear mappings to unfold the manifold. It is usually required that such a mapping preserves that local topology of the dataset. In this aim, Demartines and Herault [9] introduce an index to detect the directions in which the nonlinear projection approximatively preserves distances. Such an index can be adapted to our framework by restricting ourselves to linear projections:

$$I_{DH} \circ P_x\left(R^{k-1}\right) = \sum_{i=1}^{n}\sum_{j\neq i}\left(\left\|R_i^{k-1} - R_j^{k-1}\right\| - |P_x|(R_i^{k-1} - R_j^{k-1})\right)^2 H \circ |P_x|(R_i^{k-1} - R_j^{k-1}) \ .$$

The function H is assumed to be positive and non increasing in order to favor the local topology preservation. According the authors, the application of this function to the outputs $P_x R_i^{k-1}$ instead of the inputs R_i^{k-1} allows to obtain better performances than the Kohonen's self-organizing maps [33, 34]. Similarly, the criterion introduced in [43] yields in our case

$$I_S \circ P_x(R^{k-1}) = \sum_{i=1}^{n}\sum_{j\neq i}\left(\left\|R_i^{k-1} - R_j^{k-1}\right\| - |P_x|(R_i^{k-1} - R_j^{k-1})\right)^2 \Bigg/ \sum_{i=1}^{n}\sum_{j\neq i} P_x^2(R_i^{k-1} - R_j^{k-1}) \ .$$

However, in both cases, the resulting functions are nonlinear and thus difficult to optimize with respect to x.

Our approach is similar to Lebart one's [36]. It consists in defining a contiguity coefficient whose minimization allows to unfold nonlinear structures. At each iteration k, the following Rayleigh quotient [41] is maximized with respect to x:

$$I \circ P_x(R^{k-1}) = \sum_{i=1}^{n} P_x^2(R_i^{k-1}) \Bigg/ \sum_{i=1}^{n}\sum_{j=1}^{n} m_{i,j}^{k-1} P_x^2(R_i^{k-1} - R_j^{k-1}) \ . \quad (8.17)$$

The matrix $M^{k-1} = (m_{i,j}^{k-1})$ is a first order contiguity matrix, whose value is 1 when R_j^{k-1} is the nearest neighbor of R_i^{k-1}, 0 otherwise. The upper part of (8.17) is proportional to the usual projected variance, see (8.16). The lower part is the distance between the projection of points which are nearest neighbor in $\mathbb{R}^p$. Then, the maximization of (8.17) should reveal directions in which the projection best preserves the first order neighborhood structure (see Figure 8.1). In this sense, the index (8.17) can be seen as a first order approximation of the index proposed in [6]. Thanks to this approximation, the maximization step benefits from an explicit solution: The resulting principal direction a^k is the eigenvector associated to the maximum eigenvalue of $(V_{k-1}^{\star})^{-1}V_{k-1}$ where

$$V_{k-1}^{\star} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{i,j}^{k-1}(R_i^{k-1} - R_j^{k-1})\,{}^t(R_i^{k-1} - R_j^{k-1})$$

is proportional to the local covariance matrix. $(V_{k-1}^{\star})^{-1}$ should be read as the generalized inverse of the singular matrix $V_{k-1}^{\star}$. Indeed, since R^{k-1} is orthogonal to $\{a^1, \ldots, a^{k-1}\}$ from (8.5), $V_{k-1}^{\star}$ is, at most, of rank $p - k + 1$. Note that this approach is equivalent to Lebart's one when the contiguity matrix M is symmetric.

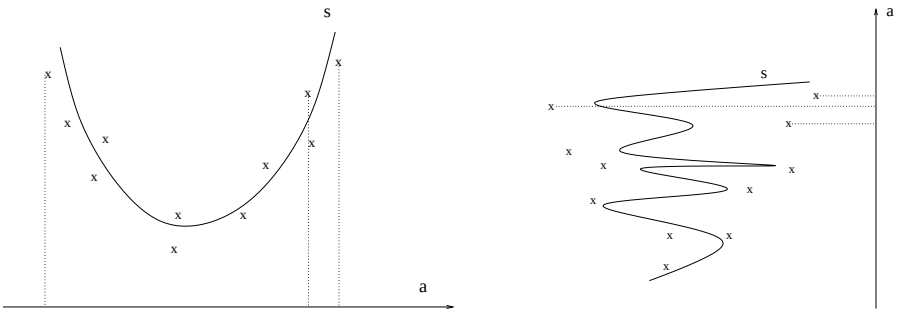

Fig. 8.1. Left: axis a such that the associated projection P_a preserves the first-order neighborhood structure. The regression function s correctly fits the dataset. Right: axis a for which P_a does not preserve the first-order neighborhood structure. The regression function s cannot yield a good approximation of the dataset

8.5 Illustration on Real and Simulated Data

Our first illustration is done on the "DistortedSShape" simulated dataset introduced in [30], paragraph 5.2.1 and available on-line[3]. The dataset consists of 100 data points in $\mathbb{R}^2$ and located around a one-dimensional curve (solid line on Figure 8.2). The bold dashed curve is the one-dimensional manifold estimated by the principal curves approach [24]. The estimated curve fails to follow the shape of the original curve. Using the auto-associative model, the estimated one-dimensional manifold (dashed curve) is closer to the original one. In this experiment, we used one iteration of Algorithm 1 with the contiguity index (8.17) in combination with a projection estimate of the regression functions. A basis of $N = 4$ cubic splines was used to compute the projection.

Our second illustration is performed on the "dataset I - Five types of breast cancer" provided to us by the organizers of the "Principal Manifolds-2006" workshop. The dataset [49] is available on-line[4]. It consists of microarray data containing logarithms of expression levels of $p = 17816$ genes in $n = 286$ samples. The data is divided into five types of breast cancer (lumA, lumB, normal, errb2 and basal) plus an unclassified group. Before all, let us note that, since n points are necessarily located on a linear subspace of dimension $n - 1$, the covariance matrix is at most of rank $n - 1 = 285$. Thus, as a preprocessing step, the dimension of the data is reduced to 285 by a classical PCA, and this, without any loss of information. Forgetting the labels, *i.e.* without using the initial classification into five types of breast cancer, the information ratio Q_d (see Corollary 1) obtained by the classical PCA and the generalized one (basing on auto-associative models), are compared. Figure 8.3 illustrates the behavior of Q_d as the dimension d of the model

[3] `http://www.iro.umontreal.ca/~kegl/research/pcurves`
[4] `http://www.ihes.fr/~zinovyev/princmanif2006/`

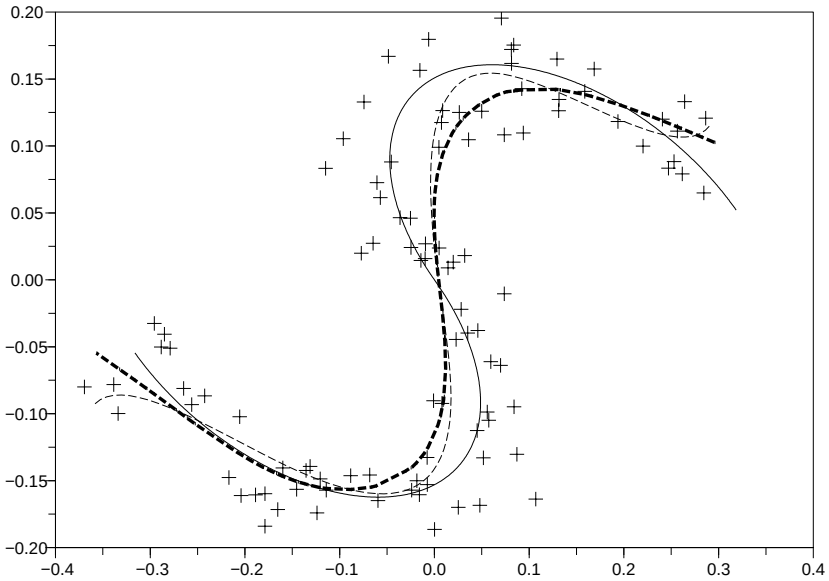

Fig. 8.2. Comparison of one-dimensional estimated manifolds on a simulated dataset. solid line: original curve, dashed line: curve estimated from the auto-associative model approach, bold dashed line: principal curve estimated by the approach proposed in [24].

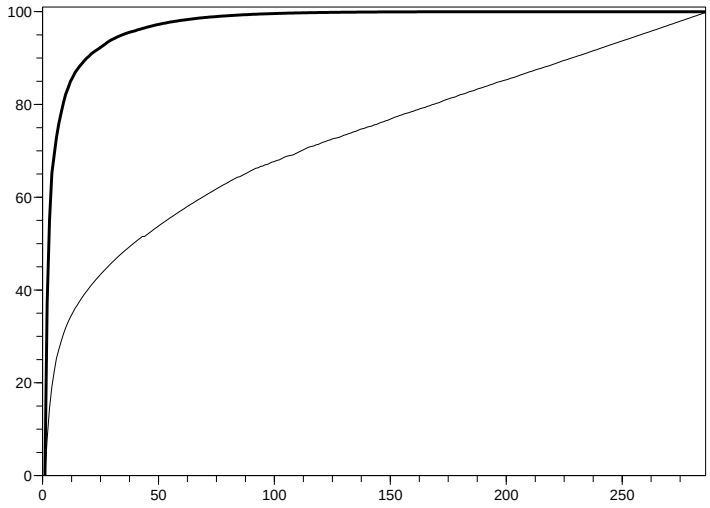

Fig. 8.3. Forgetting the labels, information ratio Q_d as a function of d on a real dataset. solid line: classical PCA, bold line: generalized PCA based on auto-associative models

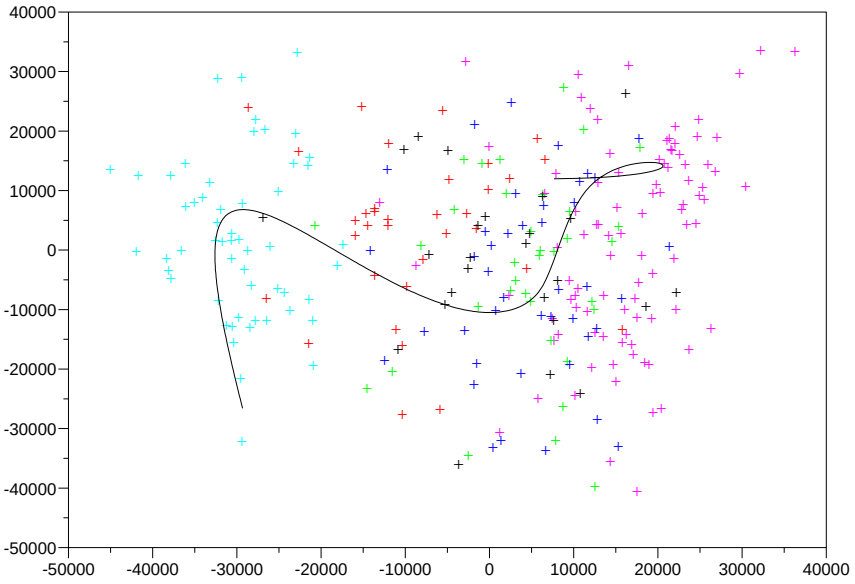

Fig. 8.4. One-dimensional manifold estimated on a real dataset with the auto-associative models approach and projected on the principal plane.

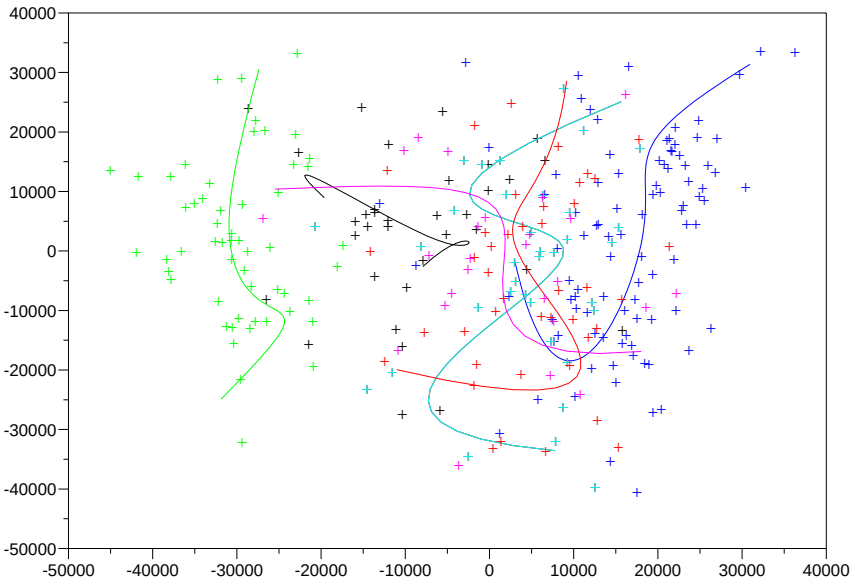

Fig. 8.5. One-dimensional manifolds estimated on each type of cancer of the real dataset with the auto-associative models approach, and projected on the principal plane

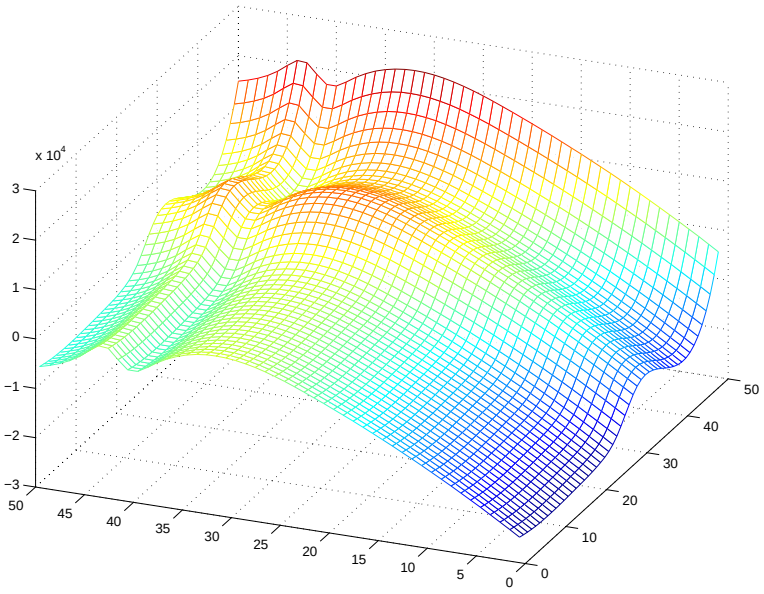

Fig. 8.6. Two-dimensional manifold estimated on a real dataset with the auto-associative models approach and projected on the three first principal axes

increases. The bold curve, corresponding to the auto-associate model, was computed with the contiguity index (8.17) in combination with a projection estimate of the regression functions. A basis of $N = 2$ cubic splines was used to compute the projection. One can see that the generalized PCA yields far better approximation results than the classical one.

As an illustration, the one-dimensional manifold is superimposed to the dataset on Figure 8.4. Each class is represented with a different gray level. For the sake of the visualization, the dataset as well as the manifold are projected on the principal plane. Similarly, the two-dimensional manifold is represented on Figure 8.6 on the linear space spanned by the three first principal axes. Taking into account the labels, it is also possible to compute the one-dimensional manifold associated to each type of cancer and to the unclassified points, see Figure 8.5. Each manifold then represents a kind of skeleton of the corresponding dataset.

Other illustrations can be found in [5], Chapter 4, where auto-associative models are applied to some image analysis problems.

References

1. Bellman, R.: Dynamic programming. Princeton university press, (1957)
2. Bosq, D. and Lecoutre, J.P.: Théorie de l'estimation fonctionnelle. Economie et Statistiques avancées. Economica, Paris (1987)
3. Carroll, J.D. and Arabie P.: Multidimensionnal scaling. Annual Rev. of Psychology, **31**, 607–649 (1980)
4. Caussinus, H. and Ruiz-Gazen, A.: Metrics for finding typical structures by means of principal component analysis. In: Data science and its Applications, pages 177–192. Harcourt Brace, Japan (1995)
5. Chalmond, B.: Modeling and inverse problems in image analysis. In: Applied Mathematics Science series, volume 155. Springer, New-York (2002)
6. Chalmond, B. and Girard, S.: Nonlinear modeling of scattered multivariate data and its application to shape change. IEEE Pattern Analysis and Machine Intelligence, **21** (5), 422–432 (1999)
7. Cheng, B. and Titterington, D.M.: Neural networks: A review from a statistical perspective. Statistical Science, **9** (1), 2–54 (1994)
8. Delicado, P.: Another look at principal curves and surfaces. Journal of Multivariate Analysis, **77**, 84–116 (2001)
9. Demartines, P. and Hérault, J.: Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. IEEE Trans. on Neural Networks, **8** (1), 148–154 (1997)
10. Diamantaras, K.L. and Kung, S.Y.: Principal component neural networks. Wiley, New-York (1996)
11. Eubank, R.L.: Spline smoothing and non-parametric regression. Decker (1990)
12. Ferraty, F. and Vieu, P.: Nonparametric modelling for functional data. Springer (2005)
13. Friedman, J.H.: Exploratory projection pursuit. Journal of the American Statistical Association, 82(397), 249–266 (1987)
14. Friedman, J.H. and Stuetzle, W.: Projection pursuit regression. Journal of the American Statistical Association, **76** (376), 817–823 (1981)
15. Friedman, J.H. and Tukey, J.W.: A projection pursuit algorithm for exploratory data analysis. IEEE Trans. on Computers, **C23** (9), 881–890 (1974)
16. Girard, S.: A nonlinear PCA based on manifold approximation. Computational Statistics, **15** (2), 145–167 (2000)
17. Girard, S., Chalmond, B., and Dinten, J.-M.: Position of principal component analysis among auto-associative composite models. Comptes-Rendus de l'Académie des Sciences, Série I, **326**, 763–768 (1998)
18. Girard, S. and Iovleff, S.: Auto-associative models and generalized principal component analysis. Journal of Multivariate Analysis, **93** (1), 21–39 (2005)
19. Gorban, A. and Zinovyev, A.: Elastic principal graphs and manifolds and their practical applications. Computing, **75** (4), 359–379 (2005)
20. Green, P.J. and Silverman, B.W.: Non-parametric regression and generalized linear models. Chapman and Hall, London (1994)
21. Hall, P.: On polynomial-based projection indices for exploratory projection pursuit. The Annals of Statistics, **17** (2), 589–605 (1990)
22. Härdle, W.: Applied nonparametric regression. Cambridge University Press, Cambridge (1990)
23. Hartigan, J.A.: Clustering algorithms. Wiley, New-York (1995)

24. Hastie, T. and Stuetzle, W.: Principal curves. Journal of the American Statistical Association, **84** (406), 502–516 (1989)
25. Hastie, T., Tibshirani, R., and Friedman, J.: The elements of statistical learning. In: Springer Series in Statistics. Springer (2001)
26. Huber, P.J.: Projection pursuit. The Annals of Statistics, **13** (2), 435–475 (1985)
27. Jolliffe, I.: Principal Component Analysis. Springer-Verlag, New York (1986)
28. Jones, M.C. and Sibson R.: What is projection pursuit? Journal of the Royal Statistical Society A, **150**, 1–36 (1987)
29. Karhunen, J. and Joutsensalo, J.: Generalizations of principal component analysis, optimization problems and neural networks. Neural Networks, **8**, 549–562 (1995)
30. Kégl, B.: "Principal curves: learning, design, and applications". PhD thesis, Concordia University, Canada (1999)
31. Kégl, B., Krzyzak, A., Linder, T., and Zeger, K.: A polygonal line algorithm for constructing principal curves. In: Proceedings of 12h NIPS, pages 501–507, Denver, Colorado (1998)
32. Klinke, S. and Grassmann, J. Projection pursuit regression. In: Wiley Series in Probability and Statistics, pages 471–496. Wiley (2000)
33. Kohonen, T.: Self-organization of topologically correct feature maps. Biological cybernetics, **43**, 135–140 (1982)
34. Kohonen, T.: Self-organization and associative memory, 3rd edition. Springer-Verlag, Berlin (1989)
35. Kruskal, J.B. and Wish, M.: Multidimensional scaling. Sage, Beverly Hills (1978)
36. Lebart L.: Contiguity analysis and classification. In: Opitz O. Gaul W. and Schader M. (eds.) Data Analysis, pages 233–244. Springer, Berlin (2000)
37. LeBlanc, M. and Tibshirani, R.: Adaptive principal surfaces. Journal of the American Statistical Association, **89** (425), 53–64 (1994)
38. Lee, J.A., Lendasse, A., and Verleysen, M.: Curvilinear distance analysis versus isomap. In: European Symposium on Artifical Neural Networks, pages 185–192. Bruges, Belgium (2002)
39. Milnor, J.: Topology from the differentiable point of view. University press of Virginia, Charlottesville (1965)
40. Pan, J-X., Fung, W-K., and Fang, K-T.: Multiple outlier detection in multivariate data using projection pursuit techniques. Journal of Statistical Planning and Inference, **83** (1), 153–167 (2000)
41. Parlett, B. N.: The symmetric eigenvalue problem. In: Classics in Applied Mathematics, vol. 20. SIAM, Philadelphia (1997)
42. Roweis, S.T. and Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science, **290**, 2323–2326 (2000)
43. Sammson, J.W.: A nonlinear mapping algorithm for data structure analysis. IEEE Transactions on Computer, **18** (5), 401–409 (1969)
44. Shepard, R.N. and Carroll, J.D.: Parametric representation of nonlinear data structures. In: Krishnaiah, P.R. (ed.) Int. Symp. on Multivariate Analysis, pages 561–592. Academic-Press (1965)
45. Tarpey, T. and Flury, B.: Self-consistency: A fundamental concept in statistics. Statistical Science, **11** (3), 229–243 (1996)
46. Tenenbaum, J.B., de Silva, V., and Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science, **290**, 2319–2323 (2000)

47. Tibshirani, R.: Principal surfaces revisited. Statistics and Computing, **2**, 183–190 (1992)
48. Vlachos, M., Domeniconi, C., Gunopulos, D., Kollios, G., and Koudas, N.: Non-linear dimensionality reduction techniques for classification and visualization. In: Proceedings of 8th SIGKDD, pages 23–26. Edmonton, Canada (2002)
49. Wang, Y., Klijn, J.G., Zhang, Y., Sieuwerts, et al.: Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. Lancet **365**, 671–679 (2005)