A nonlinear PCA based on manifold approximation

Stephane Girard

A nonlinear PCA based on manifold approximation

Stéphane Girard

Laboratoire de Probabilités et Statistique,
Université Montpellier 2, Place Eugène Bataillon,
34095 Montpellier Cedex 5, France
Email: girard@stat.math.univ-montp2.fr

Abstract

We address the problem of generalizing Principal Component Analysis (PCA) from the approximation point of view. Given a data set in a high dimensional space, PCA proposes approximations by linear subspaces. These linear models can show some limits when the data distribution is not Gaussian. To overcome these limits, we present Auto-Associative Composite (AAC) models based on manifold approximation. AAC models benefit from interesting theoretical properties, generalizing PCA ones. We take profit of these properties to propose an iterative algorithm to compute the manifold, and prove its convergence in a finite number of steps. PCA models and AAC models are first compared on a theoretical point of view. As a result, we show that PCA is the unique additive AAC model. Then a practical comparison of AAC and PCA models is presented on a data set made of curves.

1 Problem statement

Let us note $\mathcal{X} = \{x^j\}_{j=1\ldots N}$ the set of N points to approximate in $\mathbb{R}^n$. The i-th ($1 \leq i \leq n$) coordinate of point x^j ($1 \leq j \leq N$) is written x_i^j. We briefly recall the principle of PCA approximation (for further information see [26]) and exhibit its limits on a simple example. We review some contributions to overcome these limits, before presenting theoretical aspects of Auto-Associative Composite models in Section 2. Section 3 is dedicated to the implementation of these models and Section 4 presents their validation on simulations.

1.1 Principal Component Analysis

For sake of simplicity, we suppose the data to be centered. PCA builds a linear model of this set of points by approximating it with linear subspaces

$$x - \sum_{k=1}^{d} \left\langle x, a^k \right\rangle a^k = 0, \tag{1}$$

where d is the dimension of the linear subspace, and the set of axes $\{a^k\}_{k=1\ldots d}$ is an orthogonal basis of this subspace. Axes are chosen so as to minimize the mean square distance between data and the linear subspace. Consequently, PCA builds the best linear model from the $\mathcal{L}_2$ norm point of view. Moreover, if the data distribution is not Gaussian, this property remains true among the set of Auto-Associative models. (see [14] for a basic proof). Thus, it is clear that PCA limitations appear as soon as data are not Gaussian. Let us take the example of the $\mathbb{R}^3$ set of points presented in figure 1. The quality of the d-dimensional model built by PCA is measured thanks to the information ratio [22] defined by:

$$Q_d = 1 - \sum_{j=1}^{N} \left\| r^j(d) \right\|^2 / \sum_{j=1}^{N} \left\| x^j \right\|^2, \tag{2}$$

where $r^j(d) = x^j - \sum_{k=1}^{d} \left\langle x^j, a^k \right\rangle a^k$ is the approximation residual error of x^j. Whereas this set of points has an intrinsic dimension of 1, PCA

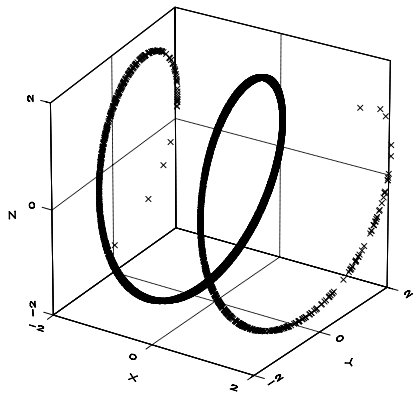


Figure 1: Example of a non Gaussian set of points.

demands to build a 3-dimensional model to get an information ratio higher than 0.75.

1.2 Some nonlinear generalizations of PCA

"Principal Curves" [17] are an intuitive PCA generalization. The idea is to replace PCA axes by curves. Since it is not possible to give the analytical expression of the curves projection operator, the model is non parametric. Besides, this approach is dedicated to the design of curves model, that is to say one dimensional manifolds.
Neural Networks can be used to build models with dimension higher than one [20], let us take Auto-Associative Perceptron example [4]. The model equation is obtained by introducing a nonlinear operator σ (called activation function) in the PCA model (1):

$$x - \sum_{k=1}^{d} \sigma^k \left(\left\langle x, a^k \right\rangle \right) = 0. \tag{3}$$

Is has been noticed [5] that such models do not lead to residual errors smaller than PCA one's because of the difficulties implied by the minimization of the model-observation distance.
Spline-PCAIV [9] and Curvilinear PCA [1] offer a different point of view by searching for transformations of the coordinates in order to maximize the information ratio of the PCA performed on the processed data.
We propose in the next section a principle to build Auto-Associative models, generalizing both Neural Networks like models and PCA models. Our models are defined in two main directions. First they are defined so as to benefit from PCA theoretical properties. Second, they offer nonlinear approximations better than PCA ones. Implementation choices are discussed in Section 3 and illustrated in Section 4 on simulated data.

2 Auto-Associative Composite models

We first recall the definition of Auto-Associative Composite models and derive some approximation properties in the second paragraph. In the third paragraph, links between PCA and Auto-Associative models are established.

2.1 Notations and Assumptions

Auto-Associative Composite models are introduced in [14] in an image analysis background. Their definition requires the introduction of projection and restoration functions.

Definition 2.1 *Given $a \in \mathbb{R}^n$, $\|a\| = 1$, define P^a the projection on the axis $[a]$ as*

$$P^a : \ x \in \mathbb{R}^n \to \langle a, x \rangle \in \mathbb{R}.$$

Definition 2.2 *Given a parameter α, define S^α the restoration function as*

$$S^\alpha : \ t \in \mathbb{R} \to S^\alpha(t) \in \mathbb{R}^n,$$

continuously differentiable and verifying
(A0) $S^\alpha(0) = 0$.

The projection-restoration function $S^{\alpha}P^{a} : \mathbb{R}^n \to \mathbb{R}^n$ is sometimes called a "bottleneck". Its use allows to overcome the curse of dimensionality [18].

Definition 2.3 *An Auto-Associative Composite (AAC) model is defined by the set of implicit equations* $\left\{G^d(\beta^d, x) = 0, d = 1, \ldots, n\right\}$*, with* [1]

$$G^d(\beta^d, x) = \left(\prod_{k=d}^{1} (Id_{\mathbb{R}^n} - S^{\alpha^k} P^{a^k})\right)(x),$$

$$\beta^d = \left\{(a^k, \alpha^k), k = 1, \ldots, d\right\},$$

under the assumptions:

(A1) $\forall k \in \{1, \ldots, d\}\ P^{a^k} S^{\alpha^k} = Id_{\mathbb{R}}$,

(A2) $\forall i, j \in \{1, \ldots, d\}\ \left\langle a^j, a^j \right\rangle = \delta_{ij}$,

(A3) $\forall i, j \in \{1, \ldots, d\}\quad i < j \Rightarrow P^{a^i} S^{\alpha^j} = 0$.

The equation $G^d(\beta^d, .) = 0$ *is called the d-th iterated model.*

To measure the goodness of fit of the d-th iterated model, we make the following definition.

Definition 2.4 *The approximation residual error of* x^j *by the d-th iterated model is noted* $r^j(d) = G(\beta, x^j)$*. The total residual error is given by*

$$R_d = \sum_{j=1}^{N} \left\| r^j(d) \right\|^2.$$

Note that $r^j(d)$ coincides with its definition paragraph 1.1 in the PCA case. We introduce below the Auto-Associative Additive model. It plays the role of an intermediary model between AAC and PCA models.

Definition 2.5 *An Auto-Associative Additive (AAA) model is a set of implicit equations* $\left\{x = F^d(\beta^d, x), d = 1, \ldots, n\right\}$*, with*

$$F^d(\beta^d, x) = \left(\sum_{k=1}^{d} S^{\alpha^k} P^{a^k}\right)(x),$$

$$\beta^d = \left\{(a^k, \alpha^k), k = 1, \ldots, d\right\},$$

such that **(A0)**-**(A3)** *hold.*

Let us stress that the Auto-Associative Perceptron (3) does not belong to the AAA class since it does not verify conditions **(A0)**-**(A3)** in the general case. This can explain the poor approximation behavior of the Perceptron, since the following section shows that assumptions **(A0)**-**(A3)** involve important consequences for the approximation properties of the model.

2.2 Approximation properties

We present how to build an AAC model iteratively. Then we show that the proposed iterative scheme consists in approximating the data set better and better by manifolds of increasing dimensions.

[1] $\prod$ stands for the operator composition, exponents stand for indices and not for powers.

2.2.1 The first-iterated model

The following lemma will prove extremely useful in the sequel. It offers a mean to build the second-iterated model starting from the first-iterated model, or more generally, to build the d-th iterated model starting from the $(d-1)$-th.

Lemma 2.1 *The residual errors are orthogonal to the first axis :*

$$\left\langle a^1, r^j(1) \right\rangle = 0, \quad \forall j \in \{1, \ldots, N\}. \tag{4}$$

Proof of Lemma 2.1: Using the definition of $r^j(1)$, il follows that

$$\left\langle a^1, r^j(1) \right\rangle = \left\langle a^1, x^j - S^{\alpha^1} P^{a^1} x^j \right\rangle = \left(Id_{\mathbb{R}} - P^{a^1} S^{\alpha^1} \right) \left(P^{a^1} x^j \right) = 0,$$

in view of **(A1)**. □

Condition **(A1)**, which is essential for this lemma, means that the restoration function S^{α^1} is a right-inverse of the projection function P^{a^1}: a scalar restoration-projection is the same scalar. The question is then to know if there exist restoration functions verifying an inverse condition, that is to say $S^{\alpha^1} P^{a^1}(x^j) = x^j$, $\forall j \in \{1, \ldots, N\}$, in order to get a perfect restoration $r^j(1) = 0$, $\forall j \in \{1, \ldots, N\}$. The answer is negative, since, in most cases the projection is non-injective on the data set $\mathcal{X}$. However, we shall see in Section 3 that it is possible to choose an axis a^1 allowing a good quality restoration $S^{\alpha^1} P^{a^1}(x^j) \simeq x^j$, $\forall j \in \{1, \ldots, N\}$.

2.2.2 The second-iterated model

The principle is the following: x^j points are projected and restored a first time, then the restoration error is measured by $r^j(1) = (Id_{\mathbb{R}^n} - S^{\alpha^1} P^{a^1})(x^j)$, and the same work is repeated on these residuals. By Lemma 2.1, they are located in the a^1 orthogonal subspace. Consequently a^2 and S^{α^2} can be chosen in the same subspace. This is why assumptions **(A2)** and **(A3)** can be made without loss of generality. Moreover, this is the basis of model properties enumerated in Theorem 1. After the second step dealing with the residuals, we have $G^2(\beta^2, x^j) = (Id_{\mathbb{R}^n} - S^{\alpha^2} P^{a^2})(r^j(1))$. Replacing $r^j(1)$, we get the equation of the second iterated model:

$$G^2(\beta^2, x) = \left(Id_{\mathbb{R}^n} - S^{\alpha^2} P^{a^2} \right) \circ \left(Id_{\mathbb{R}^n} - S^{\alpha^1} P^{a^1} \right)(x).$$

This scheme can be repeated an arbitary number of times to get the general d-th iterated model.

2.2.3 The d-th iterated model

The d-th iterated model is built iteratively with the same principle:

$$G^d(\beta^d, .) = \left(Id_{\mathbb{R}^n} - S^{\alpha^d} P^{a^d} \right) \circ G^{d-1}(\beta^{d-1}, .), \text{ with } \beta^d = \beta^{d-1} \cup (\alpha^d, a^d).$$

Data are approximated along orthogonal directions a^k, $k = 1, \ldots, d$. This spares the curse of dimensionality by only considering interesting directions. The main problem is then the choice of the axes, based on Projection Pursuit (PP) methods [11]. A function $I(a, \mathcal{X})$, called index, is maximized with respect to a. The optimization of I has been widely studied and efficient algorithms are now available to perform a Projection Pursuit [19]. Two main choices of index can be found. In the PCA case, data are assumed to be Gaussian and the index is given by $I(a, \mathcal{X}) = \mathrm{var}\left(\langle \mathcal{X}, a \rangle\right)$. At the opposite, one may choose axes along which the data distribution is non Gaussian [13] in order to see the specificity of the data set. An index appropriate to our problem is proposed in Section 3.

This kind of approach, based on an index maximization, is used in approximation problems [16] where the functions S^{α^k} are determined by the regularization functional. In Projection Pursuit Regression (PPR) algorithms [7, 12] the choice of the functions S^{α^k} is left to the user. In our case, the class of the restoration functions is a degree of freedom of the model as well. The parameters α^d of the d-th restoration function are computed by minimizing the d-th total residual error R_d:

$$\alpha^d = \arg\min_{\alpha} \sum_{j=1}^{N} \left\| r^j(d-1) - S^{\alpha} P^{a^d} r^j(d-1) \right\|^2 . \tag{5}$$

See Section 3 for the example of spline functions. Before this, let us give some properties of the model which do not depend on this choice.

Theorem 1 *AAC models share the following properties:*

1. *The d-th iterated model represents a d-dimensional manifold.*
2. *The total residual error R_d is a decreasing function of the model dimension d.*
3. *With $d = n$, the model is exact.*

In other words, the previous iterative scheme allows to approximate data better and better by manifolds of increasing dimensions. For instance, the first iterated model represents a curve without singular points. To prove Theorem 1 we need two lemmas.

Lemma 2.2 $\left\langle r^j(d+1), a^k \right\rangle = 0$, *for all* $1 \leq k \leq d+1$ *and* $1 \leq j \leq N$.

Lemma 2.3 *Gradient of the i-th coordinate of d-th iterated model can be expanded in the $\{a^1, \ldots, a^d\}$ basis as $\nabla G_i^d = a^i + U^{d,i}$ with $U^{d,i} \in [a^1, \ldots, a^d]$ for all $d+1 \leq i \leq n$.*

Proofs of these lemmas are postponed to the appendix. Let us prove now the theorem.

Proof of Theorem 1

1. We first prove that the d-th iterated model represents a d-dimensional manifold. In the $\{a^1, \ldots, a^d\}$ basis (assumption **(A2)**), the d-th iterated model is defined by a set of $(n-d)$ equations $G_i^d(\beta^d, x) = 0$, $i = d+1, \ldots, n$. A well known result [23] states that a set of equations define a d-dimensional manifold if their gradients are linearly independent. In view of Lemma 2.3, they can be written as

$$\nabla G_i^d = a^i + U^{d,i} \text{ with } U^{d,i} \in [a^1, \ldots, a^d] \quad i = d+1, \ldots, n. \tag{6}$$

Consider a linear combination of these vectors equal to zero and show that it implies that each coefficient is zero.

Let $\{\lambda_i\}_{d+1\leq i\leq n}$ be a set of scalars such as $\sum_{i=d+1}^{n} \lambda_i \nabla G_i^d = 0$. Thanks to the expansion (6), we can write the equation

$$\sum_{i=d+1}^{n} \lambda_i a^i + \sum_{i=d+1}^{n} \lambda_i U^{d,i} = 0,$$

and as $U^{d,i} \in [a^1, \ldots, a^d]$, it implies:

$$\sum_{i=d+1}^{n} \lambda_i a^i = 0.$$

Hence, the unique solution is the zero vector since the $\{a^i\}_{1\leq i\leq d}$ are linearly independent **(A2)**, and we have a d-dimensional manifold.

2. By definition (see (5)):

$$\sum_{j=1}^{N} \left\| r^j(d+1) \right\|^2 = \min_{\alpha} \sum_{j=1}^{N} \left\| r^j(d) - S^\alpha P^{a^{d+1}} r^j(d) \right\|^2.$$

Since the axes $\{a^i\}_{1\leq i\leq n}$ are orthogonal, the residual $r^j(d+1)$ can be expanded as

$$\sum_{j=1}^{N} \left\| r^j(d+1) \right\|^2 = \min_{\alpha} \sum_{j=1}^{N} \sum_{i=1}^{n} \left\| \left\langle r^j(d), a^i \right\rangle - \left(\left\langle S^\alpha, a^i \right\rangle \right) \left(P^{a^{d+1}} r^j(d) \right) \right\|^2.$$

In view of Lemma 2.1, the expansion can be limited to the $\{a^i\}_{d+1\leq i\leq n}$ axes:

$$\sum_{j=1}^{N} \left\| r^j(d+1) \right\|^2 = \min_{\alpha} \sum_{j=1}^{N} \sum_{i=d+1}^{n} \left\| \left\langle r^j(d), a^i \right\rangle - \left(\left\langle S^\alpha, a^i \right\rangle \right) \left(P^{a^{d+1}} r^j(d) \right) \right\|^2.$$

In particular, the minimum is less than the total residual obtained by choosing α^d such as $\left\langle S^\alpha, a^i \right\rangle = 0,\ i \geq d+1$, or equivalently such as $S^\alpha \in [a^1, \ldots, a^d]$.

$$\sum_{j=1}^{N} \left\| r^j(d+1) \right\|^2 \leq \sum_{j=1}^{N} \sum_{i=d+1}^{n} \left\| \left\langle r^j(d), a^i \right\rangle \right\|^2 \leq \sum_{j=1}^{N} \left\| r^j(d) \right\|^2.$$

The total residual sequence is decreasing.

3. Applying Lemma 2.2 with $d = n-1$, it follows that residuals $r^j(d)$, $j = 1, \ldots, N$ are orthogonal to a $\mathbb{R}^n$ basis (see **(A2)**). Consequently, they are zero, and the model is exact. □

We can take profit of these properties to define the information ratio of an AAC model similarly to (2).

Definition 2.6 *We define Q_d the information ratio represented by the d-th iterated model:*

$$Q_d = 1 - \sum_{j=1}^{N} \left\| r^j(d) \right\|^2 / \sum_{j=1}^{N} \left\| x^j \right\|^2 .$$

It is clear that $Q_0 = 0$. Besides, Theorem 1 shows that $(Q_d)_d$ is increasing and $Q_n = 1$. Therefore $0 \leq Q_d \leq 1$ when $0 \leq d \leq n$, and the information ratio behaves as presented in figure 2.

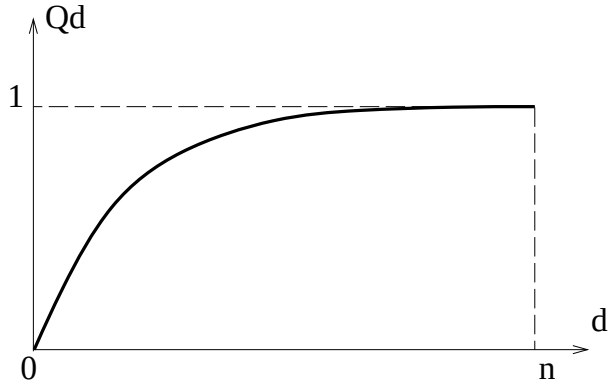


Figure 2: Behavior of the information ratio Q_d for $0 \leq d \leq n$. d stands for the dimension of the model and n for the dimension of the space.

This quantity allows to choose the model dimension according to the information ratio to represent. Let us note that these properties are the straightforward generalization of PCA ones. Comparison between PCA and AAC models is studied in the next section.

2.3 Comparison of PCA and AAC models

Theorem 2 *PCA is the only additive AAC model.*

Theorem 2 states first that PCA can be seen as a particular AAC model. This fact is not surprising since a linear subspace is also a manifold. The converse result is more interesting. Any AAA model is necessarily a PCA model. AAC models appear *a posteriori* as the more natural generalization of PCA, additivity being not possible. Theorem 2 also explains why Perceptron models (3) do not verify **(A1)**-**(A3)**.
The proof is divided into a sequence of four lemmas. Lemmas 2.4 – 2.6 are used to show that PCA is an additive AAC model, and Lemma 2.7 is used to show that it is the only one. Proofs of these Lemmas can be found in appendix or in [15].

Lemma 2.4 *PCA is a linear AAA model.*

Lemma 2.5 *(Relation between AAA and AAC models).*
Let F^d be a d-th iterated AAA model and $G^d(\beta^d, .) = Id_{\mathbb{R}^n} - F^d(\beta^d, .)$:

$$G^d(\beta^d, x) = x - \left(\sum_{k=1}^{d} S^{\alpha^k} P^{a^k} \right)(x).$$

If $P^{a^i} S^{\alpha^j} = 0$ for $1 \leq j < i \leq n$, then $G^d(\beta^d, .)$ is a d-th iterated AAC model, with the same projection and restoration functions,

$$G^d(\beta^d, x) = \left(\prod_{k=d}^{1} (Id_{\mathbb{R}^n} - S^{\alpha^k} P^{a^k}) \right)(x).$$

The proof is based on an algebraic lemma.

Lemma 2.6 *(Relations between polynomial roots and coefficients).*

$$\prod_{k=d}^{1} (X - \xi_k) = \sum_{k=0}^{d} (-1)^k \sigma_k X^{d-k}, \tag{7}$$

where σ_k is the symmetrical root function:

$$\sigma_k = \sum_{1 \leq i_1 < i_2 \cdots < i_k \leq d} \xi_{i_k} \ldots \xi_{i_2} \xi_{i_1} \text{ for } k > 0, \text{ and } \sigma_0 = 1.$$

Lemma 2.6 is used in a slightly different form to prove Lemma 2.5 in appendix.

Lemma 2.7 *If an AAC model is additive, then $P^{a^i} S^{\alpha^j} = 0$, $1 \leq j < i \leq n$.*

Proof of Theorem 1: Lemma 2.4 and Lemma 2.5 yield that PCA models are linear AAC models.
Conversely, let us consider a d-th iterated additive AAC model

$$F^d(\beta^d, x) = \sum_{k=1}^{d} S^{\alpha^k} P^{a^k}(x),$$

with the following conditions:

- $P^{a^i} S^{\alpha^j} = 0$ for $1 \leq i < j \leq n$ (assumption **(A3)**),
- $P^{a^i} S^{\alpha^j} = 0$ for $1 \leq j < i \leq n$ (Lemma 2.7).

This summarizes as $P^{a^i} S^{\alpha^j} = 0 \ \forall i \neq j$. It can be rewritten with scalar products as $\left\langle S^{\alpha^j}, a^i \right\rangle = 0 \ \forall i \neq j$. This forces $S^{\alpha^j}(t) = f^j(t) a^j$, $t \in \mathbb{R}$, using **(A2)**, for functions $f^j : \mathbb{R} \to \mathbb{R}$, $\forall j \in \{1, \ldots, d\}$. Since condition **(A1)** requires that $\left\langle S^{\alpha^j}, a^j \right\rangle = Id_{\mathbb{R}}$, it results that $f^j(t) = t$. Finally, F^d can be represented as

$$F^d(\beta^d, x) = \sum_{j=1}^{d} P^{a^j}(x) a^j.$$

This is the d-th iterated PCA model. □

3 Implementation

The algorithm to build the d-th iterated model is derived from the principle described in section 2.2.3.

3.1 Algorithm

The algorithm is presented in figure 3.1.

1. Initialization :
   $d \leftarrow 0$,
   $r^j(0) \leftarrow x^j, \quad j = 1, \ldots, N.$
2. Computation of β^{d+1}:
   - Computation of a^{d+1}:
   $$a^{d+1} = \arg\max_a I\left(a, \{r^j(d)\}_{1 \leq j \leq d}\right). \qquad (8)$$
   - Computation of α^{d+1}:
   $$\alpha^{d+1} = \arg\min_\alpha \sum_{j=1}^{N} \left\| r^j(d) - S^\alpha P^{a^{d+1}} r^j(d) \right\|^2. \qquad (9)$$
3. Residuals update:
   $r^j(d+1) \leftarrow r^j(d) - S^{\alpha^{d+1}} P^{a^{d+1}} r^j(d), \quad j = 1, \ldots, N,$
   $d \leftarrow d + 1.$
4. Go to 2. if necessary.

Figure 3: Iterative algorithm for AAC models building.

Optimizations steps (8) and (9) depend on the choices made for the projection index and for the restoration functions. Nevertheless, it is important to note that the algorithm convergence remains independent of these choices since Theorem 1 ensures us that this algorithm ends up when $d = n$, and that the information ratio increases at each iteration. In practice, the user can determine the model dimension for a given information ratio. In the next paragraph, we describe the optimization steps (8) and (9) for a particular choice of restoration functions and projection index. For sake of simplicity, the following presentation is done for the first iteration and the iteration index is given up. These simplifications can be done without loss of generality since the principle of (8) and (9) remains the same at each iteration.

3.2 Restoration functions

The class of restoration function determines the nature of the model. For instance, Lemma 2.4 asserts that choosing linear restoration function leads to a PCA model. We propose here to choose spline restoration functions, since they are proved their efficency in regression problems [10]. More generally, any set of functions used in regression frameworks could be used here (orthogonal functions, kernels ...).

Let T_M be a subdivision of an interval $[a, b]$: $T_M = \{a = t_0, \ldots, t_{M+1} = b\}$. A point $t_i, 0 \leq i \leq M+1$ is called a knot and a point $t_i, 1 \leq i \leq M$ is called an inner knot (M represents then the number on inner knots). The restoration functions are chosen as

$$S_i^{\alpha_i^k} \in s_3(T_M) \;\; 1 \leq i \leq n, \;\; 1 \leq k \leq d,$$

with $s_3(T_M)$ the set of cubic splines built on the T_M subdvision. The choice of the knots number M is crucial, it determines the dimension of the linear space $s_3(T_M)$ and the complexity of the model. It can be determined by cross-validation [28, 6] to obtain a balance between approximation and generalization. Positions of the knots are determined so as to maximize the distribution uniformity of the projected observations in the intervals of T_M. The set of parameters α is then obtained by (9) which reduces in this case to the inversion of a linear system. This estimation can also be interpretated as the building of a restoration function S^α verifying $S^\alpha P^a(x^j) \simeq x^j, \; \forall j \in \{1, \ldots, N\}$. Comparing this condition to **(A1)** $P^a S^\alpha = Id_{\mathbb{R}}$, it appears that the approximation results of the model are closely related to the choice of the projection axis. This is discussed in the next paragraph.

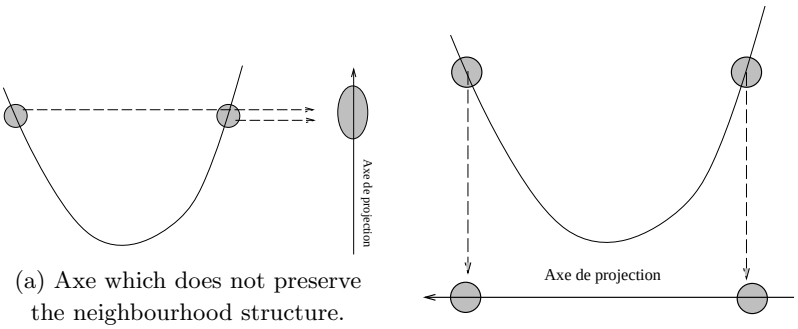
3.3 Projection index

It has already been noticed that the best case $S^\alpha P^a(x^j) = x^j, \; \forall j \in \{1, \ldots, N\}$ occurs only when the projection function is injective on the data set (figure 4b). The role of the index is then to encourage projection functions for which this condition is "almost" verified. This can be quantified by counting the number of points in the data set which are closest neighbour in $\mathbb{R}^n$ and which projections are not closest neighbour. Figure 4 illustrates this principle. In the first case (figure 4a), two points which are closest neighbour in $\mathbb{R}$ are not closest neighbour in $\mathbb{R}^n$, and the projection leads to superimpositions.

The index can be described as follows:

$$I(a, \mathcal{X}) = \sum_{i=1}^{N} \sum_{j \neq i} \Phi\left(x^j \text{ closest to } x^i\right) \Phi\left(P^a(x^j) \text{ closest to } P^a(x^i)\right).$$

Φ denotes the indicator function: $\Phi(P) = 1$ if P is true, 0 if not.



(a) Axe which does not preserve the neighbourhood structure.

(b) Axe which preserves the neighbourhood structure.

Figure 4: Choosing the projection axis.

For mathematical purposes, the index can be rewritten as

$$I(a,\mathcal{X}) = \sum_{i=1}^{N} \prod_{j>i} \left(\Phi\left[P^a\left(dx^{i\phi(i)} - dx^{ij}\right) \geq 0\right] \Phi\left[P^a\left(dx^{i\phi(i)} + dx^{ij}\right) \leq 0\right] + \Phi\left[P^a\left(dx^{i\phi(i)} - dx^{ij}\right) \leq 0\right] \Phi\left[P^a\left(dx^{i\phi(i)} + dx^{ij}\right) \geq 0\right]\right), \tag{10}$$

where $dx^{ij} = x^i - x^j$ and $x^{\phi(i)}$ represents the closest neighbour of x^i:

$$\phi(i) = \arg\min_{j \neq i} \|x^i - x^j\|.$$

The next lemma is a straightforward consequence of (10).

Lemma 3.1 *The index shares the following invariance properties :*

- $I(a, s\mathcal{X} + t) = I(a,\mathcal{X})$, $t \in \mathbb{R}^n$, $s \in \mathbb{R}$,
- $I(Da, D\mathcal{X}) = I(a,\mathcal{X})$ *with* ${}^tDD = I$.

The first invariance property with respect to translation and scale indicates that this index belongs to the class III defined by Huber [18], which is well-adapted for Projection Pursuit algorithms. The second property expresses that the search for the axis does not depend on the orientation of the data set (rotation and symmetry invariance). Similarly to the information ratio in paragraph 2.2.3, a parametrization ratio $0 \leq K_d \leq 1$ is then defined.

Definition 3.1 *We define K_d the parametrization ratio represented by the d-th axis:*

$$K_d = \frac{1}{N} I\left(a^d, \{r^j(d)\}_{1 \leq j \leq N}\right).$$

This index (10) is not continuous with respect to a. This makes any descent method useless for its maximization. To overcome this problem, we developped a simulated annealing algorithm described in [2, 3]. This method ensures to reach the global maximum of the index during the step (8) of the algorithm [8]. This whole scheme is tested on simulations in the next section.

4 Validation on simulations

4.1 A simple example

This paragraph is devoted to the illustration of the building of a 2-dimensional AAC model in $\mathbb{R}^3$. We show how residuals are reduced at each iteration by projection on linear subspaces of decreasing dimensions. Let us stress that simpler models would be much more efficient on this academic example. The data set presented in figure 5a is simulated as following:

$$\left\{ \begin{array}{lcl} x & \in & [-1,1] \\ y & \in & [0,4] \\ z & = & x^2 + \varepsilon \end{array} \right. ,$$

where ε is a centered Gaussian variable. Data are approximatively distributed on a 2-dimensional manifold. The design of the model is done according to the principle described in figure 3.1. A first projection axis a^1 is found with a parametrization ratio $K_1 = 1$. Residuals $r^j(1)$ are located in a plan orthogonal to a^1 (figure 5c) with an associated information ratio $Q_1 = 0.76$. The same approximation principle is iterated in $[a^1]^\perp$. A second projection axis a^2 is found with a parametrization ratio $K_2 = 1$. Residuals $r^j(2)$ are located on a line orthogonal to a^1 and a^2 (figure 5d), with an associated information ratio $Q_2 \simeq 1$. The algorithm is stopped and the resulting second-iterated model is presented in figure 5b.

4.2 Illustration on a data set made of curves

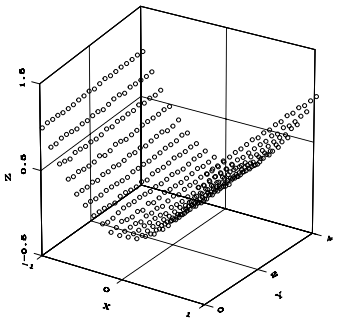
In the following example, the AAC method is tested on a data set simulated by sampling a set of N curves $\{(s, g_{tj}(s)), 1 \leq j \leq N\}$ on a fixed n-subdivision $(s_i)_{1 \leq i \leq n}$. The resulting data set writes

$$\mathcal{X} = \left\{ x^j \in \mathbb{R}^n, x_i^j = g_{tj}(s_i), 1 \leq j \leq N, 1 \leq i \leq n \right\}.$$

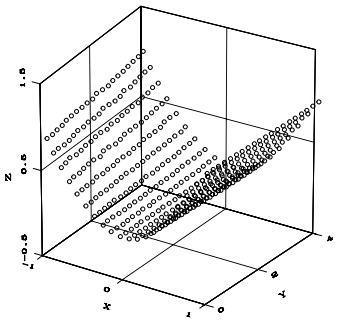
In practice we consider the case of translated curves $g_t(s) = h(t - s)$, where h is the Hanning kernel, with $n = 50$ and $N = 100$ (figure 6a). This kind of simulation is interesting for two reasons:

- Althought the data set is located in a 50-dimensional space, it can be represented in a convenient way (figure 6a).
- The data set is simulated by varying only one parameter t. This ensures that $\mathcal{X}$ is located on a 1-dimensional manifold. Projection on the principal plane illustrates this property (figure 6d).

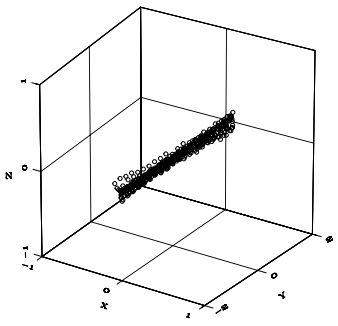
Let us note that the problem of approximating curve sets has already been addressed by Rice and Silverman [27]. They studied the effect of introducing smoothing constraints on the PCA results. This idea of functional PCA is developped in details in [24]. Besides, curve registration approaches dedicated to this problem has been proposed [21, 25]. In this paper, we focus on the comparison between PCA and AAC approximations.



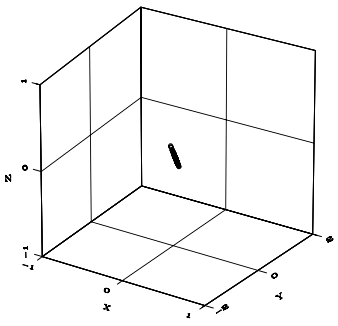
(a) Data set.



(b) Second-iterated model.

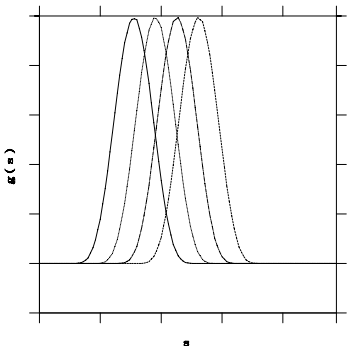


(c) First iteration residuals $r^j(1)$.

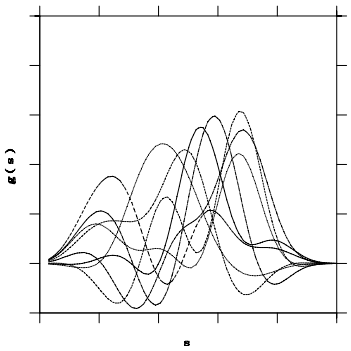


(d) Second iteration residuals $r^j(2)$.

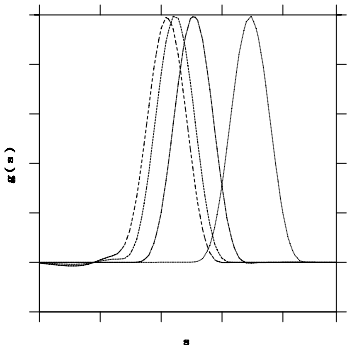
Figure 5: Simulations in $\mathbb{R}^3$.



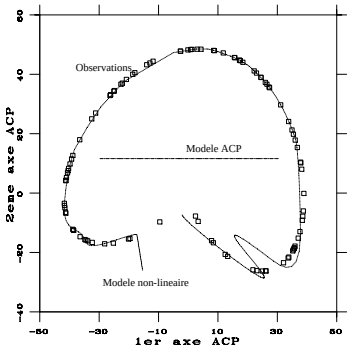
(a) A part of the data set.



(b) PCA simulations.



(c) AAC simulations.



(d) Principal plane projections.

Figure 6: Example on a data set made of curves.

4.2.1 PCA model

PCA models are ill-adapted : as the data set has not a linear structure, a 5-dimensional model has to be considered to get an information ratio Q_5 equal to 0.95. This information ratio is reached at the expense of a poor generalization behavior. Using a bootstrap technique, one can simulate new curves with the PCA model (figure 6b). It appears that PCA simulations are very far from the original data set. This phenomenon is a consequence of the PCA over-parametrization: curves very different from the original ones can be found in the 5-dimensional subspace represented by the PCA model.

4.2.2 AAC model

A projection axis is found with a parametrization ratio $K_1 = 0.93$. Since parameter t is not a linear combination of the coordinates, it is unlikely that an exact parametrization axis exists. Cross validation requires choosing $M = 18$ knots, leading to an information ratio $Q_1 = 0.95$. The corresponding manifold is projected on the principal plane for visualization (figure 6d). As the manifold remains close to the data set, simulations with the AAC models are similar to the original curves (figure 6c).

4.3 Discussion

We have seen that AAC models can be more efficient than PCA models when the data distribution is not Gaussian. However, AAC models suffer from a higher computational cost due to the optimization (8) in the step 2 of the algorithm. For instance, n evaluations of the index I (in the case of a n-dimensional model) require a number of elementary operations proportional to n^2N^2. As a comparison, the equivalent operation in the PCA case, which consists in computing the covariance matrix, only requires a number of elementary operations proportional to n^2N. Even if this difference do not prevent the use of AAC models in usual cases, it can become crucial for a very large data set. In such cases, a vector quantization algorithm can be used as a preprocessing step [29] to reduce the size of the data set.

5 Conclusion

In this paper, a nonlinear generalization of PCA is presented. Remarking that PCA approximates data with linear subspaces, we proposed a method based on manifold approximations. We show that additive models are useless in this context and so, justify the use of composite models. AAC models benefit from an efficient implementation thanks to an iterative algorithm. Some approximation properties of this model are derived and their consequences on the algorithm behavior are emphasized: reduction of the mean square error at each iteration and convergence in a finite number of steps. However, the computation of the projection axes, based on a simulated annealing procedure, is a difficult task. The subject of our current research is to find a projection index easier to maximize.

Acknowlegments

The author wishes to express his thanks to Prof. B. Chalmond and J.M. Dinten for suggesting the problem and for stimulating discussions.

References

[1] Besse, P. & Ferraty, F. (1995). A fixed effect curvilinear model. *Computational Statistics*, **10**(4), 339–351.

[2] Chalmond, B. & Girard, S. (1999). Nonlinear modeling of scattered multivariate data and its application to shape change. *IEEE Pattern Analysis and Machine Intelligence*, **21** (5), 422–431.

[3] Chalmond, B. (2000). *Eléments de modélisation pour l'analyse d'images*. Springer-Verlag, Mathmatiques et Applications 33.

[4] Cheng, B. & Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. *Statistical Science*, **9**(1), 2–54.

[5] Cottrell, M. (1994). Analyse de données et réseaux de neurones. Technical report SAMOS-Université Paris I.

[6] Craven, P. & Wahba, G. (1978). Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, **31**, 377–403.

[7] Diaconis, P. & ShahShahani, M. (1984). On nonlinear functions of linear combinations. *SIAM Journal of Scientifical Statistical Computation*, **5** (1), 175–191.

[8] Duflo, M. (1996). *Stochastic algorithms*. Springer Verlag.

[9] Durand, J. F. (1993). Generalized principal component analysis with respect to instrumental variables via univariate spline transformations. *Computational Statistics and Data Analysis*, **16**, 423–440.

[10] Eubank, R. L. (1990). *Spline smoothing and non-parametric regression*. Marcel Decker.

[11] Friedman, J. H. & Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on computers*, C**23** (9), 881–890.

[12] Friedman, J. H. & Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association*, **76** (376), 817–823.

[13] Friedman, J.H. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association*, **82** (397), 249–266.

[14] Girard, S. (1996). *Construction et apprentissage statistique de modèles auto-associatifs non-linéaires*. PhD thesis, Université de Cergy-Pontoise.

[15] Girard, S., Chalmond, B. & Dinten, J. M. (1998). Position of principal component analysis among auto-associative composite models. *Comptes-Rendus de l'Académie des Sciences*, t. **326**, Série I, 763–768.

[16] Girosi, F., Jones, M. & Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, 219–269.

[17] Hastie, T. & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, **84** (406), 502–516.

[18] Huber, P. J. (1985). Projection pursuit. *The Annals of Statistics*, **13** (2), 435–525.

[19] Huber, P. J. (1990). Algorithms for Projection Pursuit. Technical report of the Massachusetts Institute of Technology PJH-90-3.

[20] Karhunen, J. & Joutsensalo, J. (1995). Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, **8**(4), 549–562, 1995.

[21] Kneip, A. & Gasser, T. (1992). Statistical tools to analyze data representing a sample of curves. *The Annals of Statistics*, **20** (3), 1266–1305.

[22] Mardia, K.V. & Kent, J.T. & Bibby, J.M. *Multivariate Analysis*, Academic Press.

[23] Milnor, J. (1965). *Topology from the differentiable point of view*. The university press of Virginia, Charlottesville.

[24] Ramsay, J. O. & Silverman, B. W. *Functional data analysis*, Springer-Verlag, 1997.

[25] Ramsay, J. O. & Li, X. (1998). Curve Registration. *Journal of the Royal Statistical Society B*, **60** (2), 351–363.

[26] Rao, C. R. (1973). *Linear statistical inference and its application*. John Wiley and Sons.

[27] Rice, J. A. & Silverman, B. W. (1991). Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society B*, **53** (1), 233–243.

[28] Stone, M. (1974). Cross-validatory choice and assessment of statistical prediction. *Journal of the Royal Statistical Society B*, **36**, 111–147.

[29] Tzovaras, D., Strintzis, M. (1998). Use of nonlinear principal component analysis and vector quantization for image coding. *IEEE Transactions on Image Processing*, **7** (8), 1218–1223.

Appendix: Proof of Lemmas

Proof of Lemma 2.2

The proof is by induction on d. Let us note H_d the d-th hypothesis $\left\langle r^j(d), a^k \right\rangle = 0$, for $1 \leq k \leq d$ and $\forall j \in \{1, \ldots, N\}$.

- For $d = 1$, equality $\left\langle r^j(1), a^1 \right\rangle = 0,\ \forall j \in \{1, \ldots, N\}$ is a consequence of Lemma 2.1, and H_1 is true.
- Assume H_d is true and let us prove H_{d+1}.
  Let $k \in \{1, \ldots, d\}$, then for all $j \in \{1, \ldots, N\}$:

$$\begin{aligned} \left\langle r^j(d+1), a^k \right\rangle &= \left\langle r^j(d) - S^{\alpha^{d+1}} P^{a^{d+1}} r^j(d), a^k \right\rangle \\ &= \left\langle r^j(d), a^k \right\rangle - \left\langle S^{\alpha^{d+1}} P^{a^{d+1}} r^j(d), a^k \right\rangle. \end{aligned}$$

  Note $t = P^{a^{d+1}} r^j(d)$ for sake of simplicity. Then,

$$\left\langle r^j(d+1), a^k \right\rangle = \left\langle r^j(d), a^k \right\rangle - \left\langle S^{\alpha^{d+1}}(t), a^k \right\rangle,$$

  with $\left\langle r^j(d), a^k \right\rangle = 0$ in view of H_d and $\left\langle S^{\alpha^{d+1}}(t), a^k \right\rangle = 0$ by **(A3)**.
  Consequently, $\left\langle r^j(d+1), a^k \right\rangle = 0$ for $k \in \{1, \ldots, d\}$.
  Consider now the case $k = d + 1$:

$$\begin{aligned} \left\langle r^j(d+1), a^{d+1} \right\rangle &= \left\langle r^j(d), a^{d+1} \right\rangle - \left\langle S^{\alpha^{d+1}} P^{a^{d+1}} r^j(d), a^{d+1} \right\rangle \\ &= P^{a^{d+1}} r^j(d) - P^{a^{d+1}} S^{\alpha^{d+1}} P^{a^{d+1}} r^j(d) \\ &= \left(Id_{\mathbb{R}} - P^{a^{d+1}} S^{\alpha^{d+1}} \right)(t). \end{aligned}$$

  (A1) implies $P^{a^{d+1}} S^{\alpha^{d+1}} = Id_{\mathbb{R}}$ and $\left\langle r^j(d+1), a^{d+1} \right\rangle = 0$. The two previouses results prove H_{d+1}.

As a conclusion, $\left\langle r^j(d+1), a^k \right\rangle = 0,\ 1 \leq k \leq d+1$ and $\forall j \in \{1, \ldots, N\}$. □

Proof of Lemma 2.3

We show by induction on d that G_i^d gradient can be expanded as :

$$\nabla G_i^d = a^i + U^{d,i} \text{ with } U^{d,i} \in [a^1, \ldots, a^d], \quad i = d+1, \ldots, n. \quad (H_d)$$

In the orthogonal basis $\{a^1, \ldots, a^d\}$ (assumption **(A2)**), the d-th iterated model writes

$$G^d(\beta^d, x) = \left(Id_{\mathbb{R}^n} - S^{\alpha^d} P^{a^d} \right) G^{d-1}(\beta^{d-1}, x).$$

Expanding the composition product and remarking that in this basis $P^{a^d} G^{d-1}$ rewrites G_d^{d-1}, it follows

$$G^d(\beta^d, x) = G^{d-1}(\beta^{d-1}, x) - S^{\alpha^d} G_d^{d-1}(\beta^{d-1}, x).$$

The set of points represented by G^d is composed of the zeros of the $(n-d)$ following functions:

$$G_i^d(\beta^d, x) = G_i^{d-1}(\beta^{d-1,} x) - S_i^{\alpha^d} G_d^{d-1}(\beta^{d-1}, x), \quad i = d+1, \ldots, n. \quad (11)$$

- With $d = 1$, we get :

$$\nabla G_i^1 = a^i - \frac{\mathrm{d}S_i^{\alpha^1}}{\mathrm{d}t} a^1, \quad i = 2, \ldots, n,$$

  and H_1 is true.

- Assume H_{d-1} is true and let us prove H_d. Differentiating equation (11) it yields:

$$\nabla G_i^d = \nabla G_i^{d-1} - \nabla G_d^{d-1} \frac{\mathrm{d}S_i^{\alpha^d}}{\mathrm{d}t} G_d^{d-1}, \quad i = d+1, \ldots, n. \quad (12)$$

  Applying H_{d-1}, we get $\nabla G_i^{d-1} = a^i + U^{d-1,i}$ and $\nabla G_d^{d-1} = a^d + U^{d-1,d}$ with $i = d+1, \ldots, n$. Replacing in (12), ∇G_i^d rewrites

$$\nabla G_i^d = a^i + U^{d,i}, \quad i = d+1, \ldots, n,$$

  with the following definition

$$U^{d,i} = a^d + U^{d-1,i} + U^{d-1,d}.$$

  As a consequence, $U^{d,i} \in [a^1, \ldots, a^d]$ and H_d is proved. □

Proof of Lemma 2.4

Let us consider the retroprojections $S^{\alpha^k}(t) = ta^k$, $t \in \mathbb{R}$, $\alpha^k = a^k$ for $k = 1, \ldots, d$ as particular restoration functions. The d-th iterated AAA model can be written

$$F^d(\beta^d, x) = \sum_{k=1}^{d} \left\langle x, a^k \right\rangle a^k,$$

which is the d-th iterated model built by PCA. Besides, since PCA axis are orthogonal, it is straightforward to verify **(A1)**-**(A3)**. □

Proof of Lemma 2.5

Consider (7) with $X = Id_{\mathbb{R}^n}$ and $\xi_k = S^{\alpha^k} P^{a^k}$. This is possible because operators $Id_{\mathbb{R}^n}$ and $S^{\alpha^k} P^{a^k}$ commute. We have

$$\prod_{k=d}^{1} (Id_{\mathbb{R}^n} - S^{\alpha^k} P^{a^k}) = \sum_{k=0}^{d} (-1)^k Z_k,$$

with $Z_k = \sum_{1 \leq i_1 < i_2 \cdots < i_k \leq d} S^{\alpha^{i_k}} P^{a^{i_k}} \ldots S^{\alpha^{i_2}} P^{a^{i_2}} S^{\alpha^{i_1}} P^{a^{i_1}}$ for $k > 0$ and $Z_0 = Id_{\mathbb{R}^n}$. Expand the first terms of the sum:

$$\begin{aligned} \coprod_{k=d}^{1} (Id_{\mathbb{R}^n} - S^{\alpha^k} P^{a^k}) &= Id_{\mathbb{R}^n} - \sum_{k=1}^{d} S^{\alpha^k} P^{a^k} + \sum_{k=2}^{d} (-1)^k Z_k \\ &= G^d(\beta^d, .) + \sum_{k=2}^{d} (-1)^k Z_k. \end{aligned}$$

Assuming $P^{a^i} S^{\alpha^j} = 0$ for $1 \leq j < i \leq d$, and applying **(A0)**, it follows that $Z_k = 0$ for $k = 2, \ldots, d$, and $G^d(\beta^d, .)$ is the d-th iterated AAC model. □

Proof of Lemma 2.7

Les us write the i-th iterated AAC model, with $1 < i \leq n$, as

$$\begin{aligned} G^i(\beta^i, .) &= \coprod_{k=i}^{1} (Id_{\mathbb{R}^n} - S^{\alpha^k} P^{a^k}) \\ &= (Id_{\mathbb{R}^n} - S^{\alpha^i} P^{a^i}) \circ \coprod_{k=i-1}^{1} (Id_{\mathbb{R}^n} - S^{\alpha^k} P^{a^k}) \\ &= (Id_{\mathbb{R}^n} - S^{\alpha^i} P^{a^i}) \circ G^{i-1}(\beta^{i-1}, .) \\ &= (Id_{\mathbb{R}^n} - S^{\alpha^i} P^{a^i}) \circ \left(Id_{\mathbb{R}^n} - \sum_{k=1}^{i-1} S^{\alpha^k} P^{a^k} \right), \end{aligned}$$

since all the iterated models are additive by assumption. Composition products are then expanded as:

$$G^i(\beta^i, .) = Id_{\mathbb{R}^n} - \sum_{k=1}^{i} S^{\alpha^k} P^{a^k} + \sum_{k=1}^{i-1} S^{\alpha^i} P^{a^i} S^{\alpha^k} P^{a^k}.$$

Since the model is assumed to be additive, it follows that

$$\sum_{k=1}^{i-1} S^{\alpha^i} P^{a^i} S^{\alpha^k} P^{a^k} = 0.$$

This can be rewritten as

$$\forall x \in \mathbb{R}^n, \quad \sum_{k=1}^{i-1} S^{\alpha^i} P^{a^i} S^{\alpha^k} \left(\left\langle a^k, x \right\rangle \right) = 0. \tag{13}$$

In particular, taking $x = \lambda a^j$ in (13) with $\lambda \in \mathbb{R}$ and $1 \leq j \leq i-1$ yields

$$\forall \lambda \in \mathbb{R}, \quad \sum_{k \neq j} S^{\alpha^i} P^{a^i} S^{\alpha^k}(0) + S^{\alpha^i} P^{a^i} S^{\alpha^j}(\lambda) = 0, \tag{14}$$

in view of the axis orthogonality **(A2)**. Using now condition **(A0)**, (14) becomes

$$\forall \lambda \in \mathbb{R}, \quad S^{\alpha^i} P^{a^i} S^{\alpha^j}(\lambda) = 0. \tag{15}$$

By projecting (15) on axis $[a^i]$ and using condition **(A1)**, it follows that

$$\forall \lambda \in \mathbb{R}, \quad P^{a^i} S^{\alpha^j}(\lambda) = 0,$$

which is the expected result: $P^{a^i} S^{\alpha^j} = 0$ for $1 \leq j < i \leq n$. □