

Apprentissage de modèles flexibles non-linéaires pour la reconnaissance de formes

Stéphane Girard⁽¹⁾, Bernard Chalmond⁽²⁾ et Jean-Marc Dinten⁽¹⁾

⁽¹⁾ LETI CEA Technologies avancées, DSYS - CEA Grenoble
17 avenue des Martyrs, 38054 Grenoble Cedex 09, France

⁽²⁾ DIAM CMLA, Ecole Nationale Supérieure de Cachan
61, av. du président Wilson, 94235 Cachan Cedex, France

RÉSUMÉ

Nous considérons le problème de l'apprentissage de modèle pour la reconnaissance de formes du point de vue de l'analyse des données. La solution proposée est basée sur une généralisation non-linéaire de l'Analyse en Composantes Principales (ACP). Nous présentons un exemple sur la déformation de sections de fémurs.

ABSTRACT

We address the problem of model learning for pattern recognition from the multidimensional data analysis point of view. The proposed method is based on a non-linear generalization of Principal Component Analysis (PCA). We show the example of the deformations of femur cross-sections.

1 Position du problème

Nous nous intéressons au problème de la modélisation d'un objet dont la géométrie est susceptible de varier d'un exemplaire à l'autre. Il s'agit du cas d'objets manufacturés par exemple dont les dimensions sont tolérancées ou d'organes biologiques dont la géométrie est variable selon les individus. Pour prendre en compte ces variabilités, des modèles flexibles, ou déformables, doivent être considérés. Nous nous plaçons dans le cas où l'on dispose de jeux d'exemples de tels objets, supposés être représentatifs des variations géométriques pouvant intervenir dans les différentes classes. Notre optique est d'utiliser ensuite ce modèle à des fins de reconnaissance des formes : étant donné un nouvel objet, donner sa probabilité d'appartenance à une des classes d'objets du jeu d'exemples.

Dans une première partie nous décrirons l'approche couramment utilisée basée sur une Analyse en Composantes Principales (ACP) et nous montrerons ses limites. Puis nous exposerons dans une seconde partie une nouvelle méthode dont la non-linéarité permet de dépasser les limites des modèles linéaires tels que ceux issus de l'ACP. Nous montrerons ensuite quelques résultats sur des données réelles et simulées avant de conclure et de proposer quelques directions de poursuite de ces travaux.

2 Une approche classique

Nous décrivons ici une méthode fréquemment utilisée pour "apprendre" des déformations de courbes ou d'images. Nous rappellerons les hypothèses sous lesquelles la représentation obtenue est valide puis les problèmes qui apparaissent lorsque

cette hypothèse n'est plus vérifiée. Enfin, nous donnerons quelques propositions pour les résoudre trouvées dans la littérature.

2.1 Principe

Les données du jeu d'exemples sont représentées sous forme de vecteurs de points. Ces vecteurs peuvent contenir des points du contour de l'objet flexible [2], des paramètres de déformation [8] ou les ordonnées du graphe d'une fonction [5]. Dans la suite nous noterons N le nombre d'exemples et n la dimension des vecteurs. Le j ème vecteur est noté x^j et sa i ème coordonnée x_i^j . Chaque objet du jeu d'exemple est vu comme un seul point dans un espace de grande dimension : \mathbb{R}^n . L'interprétation du jeu d'exemples en tant que nuage de points est intéressante car la modélisation de déformations se ramène à un problème de statistique : l'approximation d'un nuage de points par un modèle mathématique. L'idée originale est due à Rice et Silverman [9].

Pour des raisons de simplicité de mise en oeuvre, le modèle souvent retenu est le modèle linéaire. Il est alors généralement établi (voir [4] pour une démonstration simple) que l'ACP construit le meilleur modèle linéaire du point de vue de l'approximation. Le modèle construit par l'ACP s'écrit sous sa forme paramétrée :

$$x(t^1, \dots, t^d) = \bar{x} + \sum_{k=1}^d t^k a^k \quad (1)$$

où \bar{x} est la moyenne des x^j et les a^k sont les vecteurs propres de la matrice de covariance de ces mêmes x^j (cf [10] pour plus de détails). Le modèle de l'objet est décrit comme la somme d'une forme moyenne \bar{x} et de modes de

déformations a^k dont on peut régler l'amplitude au moyen des paramètres t^k . Notons que (1) est l'équation d'un hyperplan affine. La dimension de cet hyperplan, et donc la dimension du modèle est déterminée par le nombre de paramètres d choisi. Le modèle est complété en construisant une distribution de probabilité sur ses paramètres, identifiée sur le jeu d'exemples. Le modèle étant construit, tout nouvel objet y est projeté sur celui-ci. Ses paramètres de déformation sont obtenus par produits scalaires : $\hat{t}^k = \langle y - \bar{x}, a^k \rangle$. Sa vraisemblance d'appartenance à la classe des objets en question est donnée par la vraisemblance (au sens statistique du terme) de ses paramètres \hat{t}^k .

2.2 Limitations

Comme il est montré dans [4], le modèle construit par ACP n'est valide que dans le cas de données gaussiennes. Lorsque tel n'est pas le cas, ce modèle ne permet pas de concilier approximation et généralisation. Afin d'illustrer ce phénomène considérons un exemple de déformation de courbes.

La figure 1 montre un ensemble de contours extraits de différentes coupes transverses de fémurs observées par scanner. Les fémurs présentent une légère torsion variant selon les individus [1] dont l'amplitude a été amplifiée ici par soucis de clarté des figures. Le jeu d'exemples comprend $N = 13$ courbes discrétisées en $n = 20$ points. Le nuage de points obtenu est projeté sur le plan principal (figure 2) afin de mettre en évidence son caractère non-gaussien. Les simulations obtenues avec le modèle de dimension $d = 2$ issu de l'ACP sont totalement irréalistes (figure 3). Ce dernier ne peut clairement pas être utilisé à des fins de reconnaissance de formes.

L'ACP construisant la meilleure représentation linéaire, ses limites sont inhérentes à toutes les méthodes linéaires. Des modélisations non-linéaires de différentes natures ont été proposées, nous en donnons un bref aperçu dans le paragraphe suivant.

2.3 Quelques méthodes non-linéaires

En remarquant que dans le cas de données centrées, l'équation (1) de l'hyperplan de l'ACP peut s'écrire sous forme implicite

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

différentes idées ont été proposées pour introduire des non-linéarités :

- Dans l'approche par réseaux de neurones, il a été prouvé [3] que l'équation du modèle construit par un perceptron multicouches auto-associatif s'écrit :

$$x_i - \sum_{k=1}^d B_{ik} \sigma(\langle x, A_k \rangle) = 0$$

En ayant noté A la matrice des poids des liaisons entre la couche d'entrée et la couche cachée, B la matrice des poids des liaisons entre la couche cachée et la couche de sortie et σ une fonction sigmoïdale.

- Dans l'approche de Karhunen et Joutsensalo [7] le modèle est le suivant :

$$x - \sum_{k=1}^d g(\langle x, a^k \rangle) a^k = 0$$

g est une fonction non-linéaire choisie *a priori* et les a^k sont des vecteurs de \mathbb{R}^n .

Il peut être montré [4] que pour des raisons différentes, pratiques dans le premier cas, théoriques dans le second, ces deux méthodes ne permettent pas de dépasser les limitations de l'ACP. Nous montrons dans le paragraphe suivant comment renoncer à l'additivité du modèle permet d'obtenir une méthode offrant d'importantes propriétés théoriques ainsi qu'une mise en oeuvre efficace.

3 Construction d'un modèle non-linéaire composé

Dans un premier temps nous définissons les modèles utilisés et nous exhibons quelques unes de leurs propriétés puis dans un second temps nous proposons un algorithme itératif de construction.

3.1 Définition et propriétés

Le modèle choisi est représenté sous forme implicite. Il est appelé modèle composé en raison de sa construction à partir d'un produit de composition.

Définition 3.1 — On appelle modèle non-linéaire composé tout modèle du type :

$$G(\theta, x) \stackrel{\text{def}}{=} \left(\prod_{k=d}^1 G^k(\theta^k, \cdot) \right) (x),$$

$$G^k(\theta^k, \cdot) = Id_{\mathbb{R}^n} - S^{b^k} P^{a^k}.$$

Les fonctions P^{a^k} sont des projections linéaires de \mathbb{R}^n dans \mathbb{R} : $P^{a^k}(x) = \langle a^k, x \rangle$. Les fonctions S^{b^k} représentent à l'inverse des opérateurs de restauration de \mathbb{R} dans \mathbb{R}^n . Les résultats que nous énoncerons sont indépendants du type de fonction choisi. Remarquons cependant qu'en choisissant $b^k = a^k$ et $S^{b^k}(t) = ta^k$ nous obtenons une ACP. Les paramètres a^k et b^k sont regroupés dans le vecteur θ^k .

Afin de comprendre le principe de cette modélisation, notons $r^j(d)$ le résidu d'approximation de l'observation x^j par le modèle de dimension d : $r^j(d) = G(\theta, x^j)$. On alors la relation de récurrence suivante : $r^j(d) = r^j(d-1) - S^{b^d} P^{a^d} r^j(d-1)$. En posant $r^j(0) = x^j$, il apparaît que le modèle est tout naturellement construit en approchant itérativement les différents résidus par une série de compressions-restaurations. Le lemme suivant précise la façon d'itérer la construction du modèle.

Lemme 3.2 — Si la compression-restauration est la fonction identité,

$$P^{a^k} S^{b^k} = Id_{\mathbb{R}}, \quad (2)$$

alors les résidus $r^j(k)$ sont perpendiculaires à l'axe a^k :

$$P^{a^k}(r^j(k)) = 0, \quad \forall j.$$

La preuve est très simple :

$$\begin{aligned} \text{Démonstration : } P^{a^k}(r^j(k)) &= P^{a^k}[x^j - S^{b^k} P^{a^k}(x^j)] = \\ P^{a^k}(x^j) - P^{a^k} S^{b^k} P^{a^k}(x^j) &= P^{a^k}(x^j) - Id_{\mathbb{R}} P^{a^k}(x^j) = 0 \quad \blacksquare \end{aligned}$$

Ce lemme est d'une grande importance pratique : une fois le k ème terme $G^k(\theta^k, \cdot)$ du modèle construit, les résidus se trouvent dans l'orthogonal au k ème axe a^k . La construction du terme suivant peut donc se faire de la même façon que précédemment dans ce sous-espace vectoriel. L'algorithme sera décrit plus précisément dans le paragraphe suivant. Remarquons que la condition (2) est naturelle : tout point sur l'axe une fois restauré doit voir sa projection confondue avec lui-même. Ce lemme est également à la base des propriétés suivantes :

Proposition 3.3 — *Les modèles composés ont les propriétés suivantes :*

- Le modèle $G(\theta, x) = 0$ avec d axes définit une variété de dimension d .
- Les résidus d'approximation moyens sont décroissants.
- Avec $d = n$, le modèle est exact.

Démonstration : Ces propriétés sont démontrées dans [4]. Remarquons tout de même que la dernière est une conséquence de la première. \blacksquare

La première propriété précise la nature du modèle. La variété construite est l'extension des sous-espace affines de l'ACP. Les deux dernières propriétés sont directement généralisées de celles de l'ACP. Elles permettent d'affirmer que l'algorithme de construction du modèle présenté ci-dessous converge vers la solution exact en un nombre fini d'itérations.

3.2 Algorithme

1. Initialisation :
 $k \leftarrow 0$
 $r^j(0) \leftarrow x^j, j = 1, \dots, N$
2. Calcul des paramètres $\theta^{k+1} = (a^{k+1}, b^{k+1})$:
 – Calcul de l'axe a^{k+1} .

$$a^{k+1} = \arg \max_a I(a, \{r^j(k)\}_{j=1}^N) \quad (3)$$

 – Calcul du paramètre b^{k+1} .

$$b^{k+1} = \arg \min_b \sum_{j=1}^N \|r^j(k) - S^b P^{a^{k+1}}(r^j(k))\|^2 \quad (4)$$
3. Mise à jour des résidus.
 $r^j(k+1) \leftarrow r^j(k) - S^{b^{k+1}} P^{a^{k+1}}(r^j(k)), j = 1, \dots, N$
4. Retour en 2, si les résidus sont trop importants, avec $k \leftarrow k + 1$; ou sinon $d = k$ et fin.

Afin d'implanter cet algorithme il est nécessaire d'en préciser la seconde étape. Les axes de projection sont choisis par maximisation d'un index de projection (3). Il s'agit ici d'une technique de Poursuite de Projection [6]. L'index I choisi permet de qualifier la qualité d'un axe a en terme de conservation locale de la structure de voisinage par la projection P^a (cf [4]). D'autre part la minimisation (4) dépend directement de la forme des fonctions de lissage. Nous avons choisi dans d'utiliser des splines cubiques (les b^k représentent alors leurs

coefficients) d'une grande efficacité dans les problèmes de régression. Des résultats sont montrés sur l'exemple vu précédemment.

4 Résultats

La méthode de construction d'un modèle non-linéaire est appliquée sur l'exemple présenté en introduction. Une variété de dimension $d = 1$ est construite grâce à une itération de l'algorithme ci-dessus. On peut voir une projection de cette variété sur le plan principal (figure 2) et constater que celle-ci reste très proche des données de jeu d'exemples. Une distribution de probabilité de type "mélange de gaussiennes" est construite sur le paramètre du modèle. Par tirage aléatoire, on simule alors des réalisations (figure 4) beaucoup plus cohérentes avec les données initiales (figure 1) que les simulations effectuées par ACP (figure 3).

5 Conclusion

Nous avons présenté une nouvelle méthode de construction et d'apprentissage de modèles pour la reconnaissance des formes. Le problème est considéré du point de vue de l'analyse des données ce qui permet d'obtenir une méthode générique indépendante de l'application envisagée. Les performances de cette méthode dépassent largement celles de l'ACP limitées aux données gaussiennes.

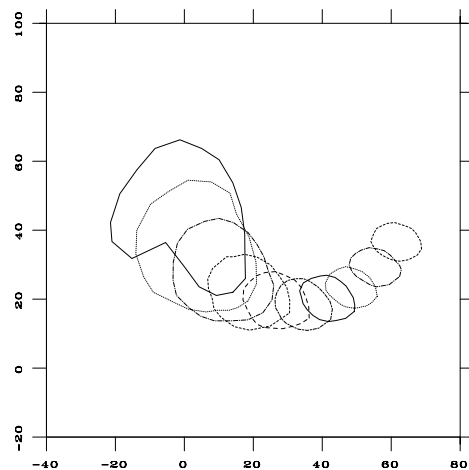


FIG. 1 — Jeu d'exemples

Références

- [1] B. Chalmond. Individual hip prosthesis design from ct images. *Pattern Recognition Letters*, 8 :203–208, 1988.
- [2] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J.Graham. Training models of shape from sets of examples. In *3rd british machine vision conference*, pages 9–18, Leeds, 1992.

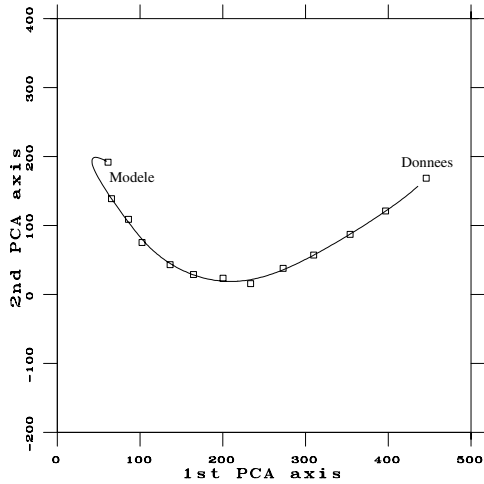


FIG. 2 — Projections sur le plan principal

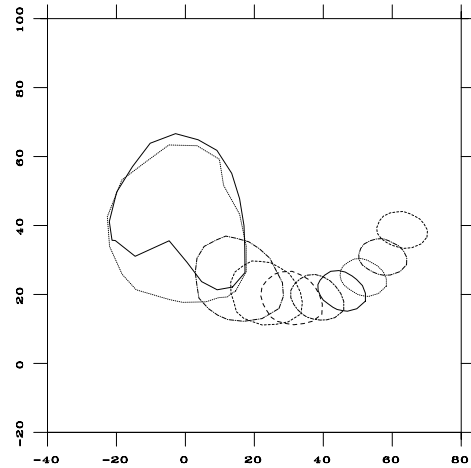


FIG. 4 — Simulations "non-linéaires"

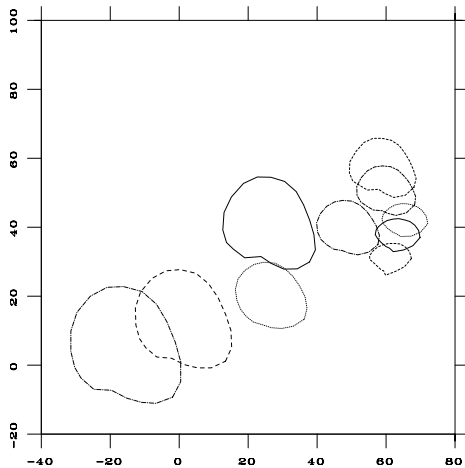


FIG. 3 — Simulations par ACP

- [3] M. Cottrell. Analyse de données et réseaux de neurones. préprint SAMOS-Université Paris I, février 1994.
- [4] S. Girard. *Construction et apprentissage statistique de modèles auto-associatifs non-linéaires*. PhD thesis, Université de Cergy-Pontoise, Octobre 1996.
- [5] S. Girard, J.M. Dinten, and B. Chalmond. Building and training radiographic flexible prior models for object identification from incomplete data. *IEE proceedings on Vision, Image and Signal Processing*, 143(4) :257–264, 1996.
- [6] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13 (2) :435–525, 1985.
- [7] J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4) :549–562, 1995.
- [8] C. Nastar, B. Moghaddam, and A. Pentland. Flexible images : matching and recognition using learned deformations. *CVGIP : Image Understanding*, 65 (2) :179–191, 1997.

- [9] J.A. Rice and B.W. Silverman. Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal Royal Statistical Society B*, 53 (1) :233–243, 1991.
- [10] G. Saporta. *Probabilités, analyse des données et statistique*. Editions Technip, Paris, 1990.