Analyse en Composantes Principales de données floues par réseau de neurones auto-associatif

Principal component analysis of fuzzy data using autoassociative neural networks

M. Masson T. Denœux

Heuristique et Diagnostic des Systèmes Complexes – UMR CNRS 6599
Université de Technologie de Compiègne
BP 529 - F-60205 Compiègne cedex - France

Résumé

Cet article décrit une extension de la méthode d'analyse en composantes principales, permettant d'extraire un nombre limité de caractéristiques à partir de données floues multidimensionnelles. Notre approche se fonde sur des résultats récents concernant la capacité de réseaux de neurones linéaires autoassociatifs à réaliser une compression de l'information d'entrée, tout comme l'ACP, sans diagonalisation explicite de la matrice variance-covariance. Les expériences menées sur des données simulées et des données d'évaluation sensorielle ont montré la capacité de la méthode à fournir des représentations synthétiques de données floues multidimensionnelles.

Mots Clef

Analyse de données floues, extraction de caractéristiques, réseaux de neurones, reconnaissance de formes.

Abstract

This paper describes an extension of principal component analysis allowing to extract a limited number of relevant features from high dimensional fuzzy data. Our approach is based on recent results concerning the ability of linear autoassociative neural networks to perform information compression without any explicit matrix diagonalization. Experiments with artificial and real sensory evaluation data demonstrate the ability of the method to provide concise representations of complex fuzzy data.

Keywords

Fuzzy data analysis, feature extraction, neural networks, pattern recognition.

1 Introduction

L'un des objectifs de l'analyse de données exploratoire est de fournir une représentation à la fois fidèle et synthétique de données multidimensionnelles. Une des méthodes les plus simples et les plus utilisées dans ce cadre est l'Analyse en Composantes Principales (ACP). Etant donné un ensemble de n vecteurs multidimensionnels, $\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$, de dimension d, l'ACP consiste à rechercher des directions dans l'espace le long desquelles la variance des points est maximale. Si q directions sont trouvées (avec $q < d$), celles-ci définissent un sous-espace linéaire $\mathcal{L}$ de $\mathbb{R}^d$ tel que les projections $\{\mathbf{y}^1, \ldots, \mathbf{y}^n\}$ des vecteurs initiaux sur $\mathcal{L}$, ayant une dispersion maximale, constituent en un certain sens un résumé optimal des données initiales. Cette méthode est très employée pour étudier la structure de données multidimensionnelles grâce à des représentations planes des données, ou encore comme procédure d'extraction de caractéristiques avant une phase de classification ou de régression (cf [7]). Dans ce qui suit, il est proposé d'étendre l'ACP à une classe plus large de données comprenant des données de type intervalle ou, plus généralement, des nombres flous. Les nombres flous sont définis comme des ensembles flous de l'axe des réels, avec une fonction d'appartenance quasi-concave et semi-continue supérieurement [6]. Ils peuvent permettre de modéliser des observations vagues ou imprécises, ou encore des distributions de valeurs. Dans la suite de cet article, nous considérerons que les données consistent en un ensemble $\{\widetilde{\mathbf{x}}^1, \ldots, \widetilde{\mathbf{x}}^n\}$ de n vecteurs de dimension d de nombres flous. Notre but est de définir une "ACP généralisée" pour obtenir une représentation synthétique sous la forme de n vecteurs de dimension q de nombres flous $\{\widetilde{\mathbf{y}}^1, \ldots, \widetilde{\mathbf{y}}^n\}$. L'approche proposée est fondée sur une implémentation particulière de l'ACP : un réseau de neurones autoassociatif composé de trois couches. La couche d'entrée et la couche de sortie comportent d cellules, tandis que la couche centrale (dite couche cachée) comporte q cellules. Par minimisation d'un critère adéquat, on obtient en sortie de la couche cachée une représentation répondant aux objectifs fixés. Tandis que de nombreux travaux ont porté sur le developpement d'algorithmes flous pour l'analyse de données "nettes" (cf par exemple [3]), on trouve peu d'études portant sur l'analyse exploratoire de données floues. Bien qu'elles ne traitent pas du problème de visualisation de données multidimensionnelles qui nous concerne, [5, 8, 9, 11] sont quelques references récentes dans ce domaine. L'article est organisé comme suit : le paragraphe 2 rappelle certains points relatifs à l'ACP et à son implémentation par réseau autoassociatif. Le paragraphe 3 est consacré à la présentation de la méthode proposée. Enfin, nous présentons quelques résultats expérimentaux.

2 ACP par réseaux de neurones

2.1 Rappels sur l'ACP classique

Soit un ensemble de données constitué de n vecteurs réels $\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$ de dimension d. En toute généralité, on supposera que les données sont centrées :

$$\frac{1}{n}\sum_{p=1}^{n} \mathbf{x}^p = 0.$$

Avec cette hypothèse, on peut voir cet ensemble de données comme un nuage de points dans un espace euclidien centré sur l'origine. L'ACP consiste à chercher un sous-espace $\mathcal{L}$ de dimension q de telle sorte que les projections orthogonales $P_{\mathcal{L}}\mathbf{x}^p$ des n points sur $\mathcal{L}$ ont une variance maximale. On peut montrer que la solution de ce problème est donnée par le sous-espace engendré par les q premiers vecteurs propres $\mathbf{u}_j$ de la matrice S de variance-covariance des données :

$$S = \frac{1}{n}\sum_{p=1}^{n} \mathbf{x}^p \mathbf{x}^{p'}.$$

De manière équivalente, on peut définir le sous-espace optimal $\mathcal{L}$ comme celui pour lequel l'erreur de projection moyenne

$$\frac{1}{n}\sum_{p=1}^{n} \|\mathbf{x}^p - P_{\mathcal{L}}\mathbf{x}^p\|^2$$

est minimale. Les vecteurs $\mathbf{y}^p$ peuvent être calculés sous forme matricielle comme $\mathbf{y}^p = U_q'\mathbf{x}^p$, où $U_q = [\mathbf{u}_1, \ldots, \mathbf{u}_q]$ est la matrice de dimension $d \times q$ composée en colonnes des q vecteurs propres normalisés de S.

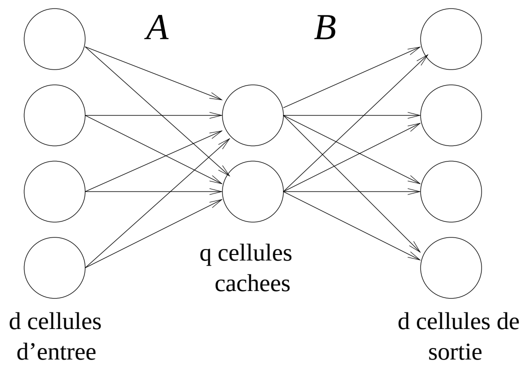

FIG. 1 – Réseau de neurones autoassociatif.

2.2 Réseaux auto-associatifs linéaires

Comme l'ont noté Bourlard et Kamp [4], il existe un lien intéressant entre l'ACP et les perceptrons autoassociatifs. Considérons un réseau multicouches, représenté figure 1, comportant une couche d'entrée et une couche de sortie à d cellules linéaires, et une couche cachée à q cellules linéaires.
Soient A la matrice de dimension $q \times d$ des poids de la couche d'entrée vers la couche cachée, et B la matrice de dimension $d \times q$ des poids de la couche cachée vers la couche de sortie. La sortie du réseau, pour une entrée $\mathbf{x}$, se calcule comme

$$\mathbf{z} = BA\mathbf{x} = W\mathbf{x}, \tag{1}$$

où W désigne la transformation globale de $\mathbf{x}$ en $\mathbf{z}$. On suppose que le réseau est entraîné en mode autoassociatif, c'est à dire que les entrées sont les sorties désirées du réseau. Si $q < d$, une telle procédure, en cherchant W aussi proche que possible de l'identité, oblige le système à trouver une manière efficace de compresser l'information contenue dans les données. Baldi et Hornik [1, 2] ont pu montrer que si l'on minimise le critère suivant :

$$E(A, B) = \sum_{p=1}^{n} e(\mathbf{x}^p, \mathbf{z}^p), \tag{2}$$

où $e(\mathbf{x}^p, \mathbf{z}^p)$ représente l'erreur de reconstruction pour la forme p :

$$e(\mathbf{x}^p, \mathbf{z}^p) = \|\mathbf{x}^p - \mathbf{z}^p\|^2 = \sum_{k=1}^{d}(x_k^p - z_k^p)^2, \quad p = 1, \ldots, n. \tag{3}$$

alors, la solution optimale pour les matrices A et B est de la forme :

$$\begin{aligned} A &= CU'_q \quad (4) \\ B &= U_q C^{-1}, \quad (5) \end{aligned}$$

où U_q est la matrice des axes principaux $[\mathbf{u}_1, \ldots, \mathbf{u}_q]$, et C est une matrice inversible arbitraire de dimension $q \times q$.

On retrouve donc l'ACP comme correspondant au cas particulier où C est égale à l'identité. Les sorties de la couche cachée sont dans ce cas identiques aux composantes principales des données. Cette solution n'est cependant pas systématiquement obtenue, et les sorties de la couche cachée sont les composantes principales définies à une transformation linéaire près. Une façon de contourner la difficulté est d'introduire la contrainte $A' = B$. Cette condition impose que $CC' = I$, c'est-à-dire que C soit une matrice orthogonale. Dans ce cas, les composantes principales sont les sorties de la couche cachée à une transformation *isométrique* près (symétrie et rotation), ce qui est plus satisfaisant en termes de pouvoir de représentation des caractéristiques extraites. L'équation (1) se réécrit dans ce cas sous la forme suivante :

$$\mathbf{z} = BB'\mathbf{x} \quad (6)$$

ou encore, sous forme scalaire,

$$z_k = \sum_{j=1}^{q} B_{kj} \sum_{i=1}^{d} B_{ij} x_i \quad k = 1, \ldots, d. \quad (7)$$

En pratique, les poids du réseau (la matrice B) sont obtenus en minimisant l'erreur de reconstruction moyenne par une technique classique de descente de gradient.

3 Extension à des données floues

3.1 Principe

Soit $\mathcal{F}(\mathbb{R})$ l'ensemble des nombres flous définis sur $\mathbb{R}$. On suppose maintenant que les données consistent en un ensemble de vecteurs $\{\widetilde{\mathbf{x}}^1, \ldots, \widetilde{\mathbf{x}}^n\}$, où chaque $\widetilde{\mathbf{x}}^p \in \mathcal{F}(\mathbb{R})^d$ est un vecteur de d nombres flous notés $(\widetilde{x}_i^p)_{1 \leq i \leq d}$. Le but est de compresser ces données sous la forme de vecteurs $\{\widetilde{\mathbf{y}}^1, \ldots, \widetilde{\mathbf{y}}^n\}$ de nombres flous de dimension inférieur, avec $\widetilde{\mathbf{y}}^p \in \mathcal{F}(\mathbb{R})^q$, $p = 1, \ldots, n$, et $q < d$. Dans le cas de données classiques, ce problème peut être résolu par une simple ACP. L'implémentation de celle-ci par réseau autoassociatif, décrite au paragraphe précédent, donne une solution pour la généralisation au cas flou. On considère un réseau dont la structure est la même que précédemment et l'on suppose qu'un vecteur $\widetilde{\mathbf{x}}$ de d nombres flous est placé en entrée du réseau. La sortie du réseau peut être calculée en appliquant le principe d'extension de Zadeh à l'équation (6). La $k^{\text{ième}}$ composante $\widetilde{z}_k$ du vecteur flou de sortie $\widetilde{\mathbf{z}}$ pour l'entrée $\widetilde{\mathbf{x}}$ est donc définie comme :

$$\mu_{\widetilde{z}_k}(u) = \sup \min_{1 \leq i \leq d} \mu_{\widetilde{x}_i}(v_i), \quad (8)$$

le supremum étant pris sous la contrainte :

$$u = \sum_{j=1}^{q} B_{kj} \sum_{i=1}^{d} B_{ij} v_i.$$

Une forme plus compacte est donnée par :

$$\widetilde{z}_k = \sum_{j=1}^{q} B_{kj} \sum_{i=1}^{d} B_{ij} \widetilde{x}_i \quad k = 1, \ldots, d, \quad (9)$$

où l'addition et la multiplication par un réel sont remplacées par les opérations usuelles d'arithmétique floue [6], ou encore, sous une forme analogue à celle de l'équation (6) :

$$\widetilde{\mathbf{z}} = BB'\widetilde{\mathbf{x}}. \quad (10)$$

Si le réseau est correctement entraîné, le vecteur $\widetilde{\mathbf{y}} = B'\widetilde{\mathbf{x}}$ constitue une version compressée de la donnée floue d'entrée $\widetilde{\mathbf{x}}$. Il s'agit donc de déterminer comment les poids du réseau (la matrice B) peuvent être optimisés et de généraliser le critère d'erreur défini par l'équation (2). Dans le paragraphe suivant, nous présentons une manière de procéder, adaptée au cas de nombres flous trapézoïdaux. La démarche reste néanmoins plus générale.

3.2 Cas de nombres flous trapézoïdaux

Les nombres flous trapézoïdaux sont utilisés fréquemment pour la modélisation de données imprécises pour des raisons de simplicité de manipulation et d'efficacité calculatoire. Nous en rappelons ici les principales caractéristiques : soient $x^{(1)}, x^{(2)}, x^{(3)}$ and $x^{(4)}$ quatre nombres réels tels que $x^{(1)} \leq x^{(2)} \leq x^{(3)} \leq x^{(4)}$. La quantité $\widetilde{x}$ définie par la fonction d'appartenance suivante :

$$\mu_{\widetilde{x}}(u) = \begin{cases} 0 & \text{if } u \leq x^{(1)} \text{ or } u \geq x^{(4)} \\ \dfrac{u - x^{(1)}}{x^{(2)} - x^{(1)}} & \text{if } x^{(1)} \leq u \leq x^{(2)} \\ 1 & \text{if } x^{(2)} \leq u \leq x^{(3)} \\ \dfrac{u - x^{(3)}}{x^{(4)} - x^{(3)}} & \text{if } x^{(3)} \leq u \leq x^{(4)} \end{cases} \quad (11)$$

est appelée un nombre flou trapézoïdal et notée $\widetilde{x} = (x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$. Les nombres flous peuvent être utilisés aussi bien pour modéliser des données précises qu'imprécises. Par exemple, un nombre réel x se représente par (x, x, x, x), un intervalle $[a, b]$ par (a, a, b, b). Ils offrent donc un cadre souple pour la fusion d'informations hétérogènes. Outre cette caractéristique, les nombres flous trapézoïdaux possèdent l'attrait de former une famille stable vis à vis des opérations d'addition, de soustraction et de multiplication par un réel. De façon plus précise, soit $\widetilde{x} = (x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$ et $\widetilde{y} = (y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)})$ deux nombres flous trapézoïdaux. Alors, on peut montrer [6] que :

$$\widetilde{x} + \widetilde{y} = (x^{(1)} + y^{(1)}, x^{(2)} + y^{(2)}, x^{(3)} + y^{(3)}, x^{(4)} + y^{(4)}) \tag{12}$$

$$\widetilde{x} - \widetilde{y} = (x^{(1)} - y^{(4)}, x^{(2)} - y^{(3)}, x^{(3)} + y^{(2)}, x^{(4)} - y^{(1)}) \tag{13}$$

$$\forall a \in \mathbb{R}, \quad a\widetilde{x} = \begin{cases} (ax^{(1)}, ax^{(2)}, ax^{(3)}, ax^{(4)}) & \text{if } a \geq 0 \\ (ax^{(4)}, ax^{(3)}, ax^{(2)}, ax^{(1)}) & \text{if } a < 0 \end{cases} \tag{14}$$

On considère donc dans ce qui suit que les composantes $\widetilde{x}_i$ du vecteur d'entrée $\widetilde{\mathbf{x}} \in \mathcal{F}(\mathbb{R})^d$ sont des nombres flous trapézoïdaux $\widetilde{x}_i = (x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, x_i^{(4)})$. De part les propriétés évoquées précédemment, les sorties du réseau sont également des nombres flous trapézoïdaux. On calcule dans un premier temps les sorties de la couche cachée. Par définition, la sortie $\widetilde{y}_j$ de la $j^{\text{ième}}$ cellule cachée s'écrit :

$$\widetilde{y}_j = \sum_{i=1}^{d} B_{ij}\widetilde{x}_i = (y_j^{(1)}, y_j^{(2)}, y_j^{(3)}, y_j^{(4)}) \quad j = 1, \ldots, q. \tag{15}$$

En utlisant les propriétés (12-14), il vient :

$$y_j^{(1)} = \sum_{\substack{i=1 \\ B_{ij}>0}}^{d} B_{ij}x_i^{(1)} + \sum_{\substack{i=1 \\ B_{ij}<0}}^{d} B_{ij}x_i^{(4)}, \tag{16}$$

$$y_j^{(2)} = \sum_{\substack{i=1 \\ B_{ij}>0}}^{d} B_{ij}x_i^{(2)} + \sum_{\substack{i=1 \\ B_{ij}<0}}^{d} B_{ij}x_i^{(3)}, \tag{17}$$

$$y_j^{(3)} = \sum_{\substack{i=1 \\ B_{ij}>0}}^{d} B_{ij}x_i^{(3)} + \sum_{\substack{i=1 \\ B_{ij}<0}}^{d} B_{ij}x_i^{(2)}, \tag{18}$$

$$y_j^{(4)} = \sum_{\substack{i=1 \\ B_{ij}>0}}^{d} B_{ij}x_i^{(4)} + \sum_{\substack{i=1 \\ B_{ij}<0}}^{d} B_{ij}x_i^{(1)}. \tag{19}$$

De même, la $k^{\text{ième}}$ sortie est par définition :

$$\widetilde{z}_k = \sum_{j=1}^{p} B_{kj}\widetilde{y}_j = (z_k^{(1)}, z_k^{(2)}, z_k^{(3)}, z_k^{(4)}) \quad k = 1, \ldots, d, \tag{20}$$

et donc :

$$z_k^{(1)} = \sum_{\substack{k=1 \\ B_{kj}>0}}^{q} B_{kj}y_j^{(1)} + \sum_{\substack{k=1 \\ B_{kj}<0}}^{q} B_{kj}y_j^{(4)}, \tag{21}$$

$$z_k^{(2)} = \sum_{\substack{k=1 \\ B_{kj}>0}}^{q} B_{kj}y_j^{(2)} + \sum_{\substack{k=1 \\ B_{kj}<0}}^{q} B_{kj}y_j^{(3)}, \tag{22}$$

$$z_k^{(3)} = \sum_{\substack{k=1 \\ B_{kj}>0}}^{q} B_{kj}y_j^{(3)} + \sum_{\substack{k=1 \\ B_{kj}<0}}^{q} B_{kj}y_j^{(2)}, \tag{23}$$

$$z_k^{(4)} = \sum_{\substack{k=1 \\ B_{kj}>0}}^{q} B_{kj}y_j^{(4)} + \sum_{\substack{k=1 \\ B_{kj}<0}}^{q} B_{kj}y_j^{(1)}. \tag{24}$$

Un expression analytique de la fonction d'erreur à optimiser peut être obtenue en choisissant une métrique adaptée. En considérant chaque nombre flou trapézoïdal comme un point dans un espace à quatre dimensions comme proposé dans [10] et [5], on peut généraliser la fonction d'erreur (2) par le critère suivant :

$$E(B) = \sum_{p=1}^{n}\sum_{k=1}^{d} e(\widetilde{x}_k^p, \widetilde{z}_k^p)$$

avec,

$$e(\widetilde{x}_k, \widetilde{z}_k) = \sum_{t=1}^{4}(z_k^{(t)} - x_k^{(t)})^2, \quad k = 1, \ldots, d. \tag{25}$$

Les expressions explicites des dérivées partielles de $E(B)$ par rapport aux poids B_{ij} du réseau s'obtiennent alors facilement en notant que :

$$\frac{\partial e(\widetilde{x}_k^p, \widetilde{z}_k^p)}{\partial B_{ij}} = \sum_{t=1}^{4} 2(z_k^{(t)} - x_k^{(t)})\frac{\partial z_k^{(t)}}{\partial B_{ij}} \tag{26}$$

et en remplaçant les dérivées partielles de $\partial z_k^{(t)}/\partial B_{ij}$ par leur expression fournie en annexe (tableau 5).

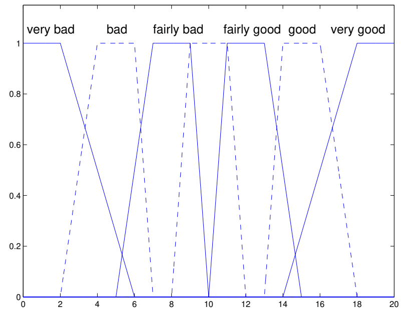

FIG. 2 – Fonctions d'appartenance des notes floues.

4 Exemples d'application

4.1 Données simulées

Pour illustrer la capacité de la méthode à fournir une représentation condensée adéquate des données, nous reprenons un exemple adapté de [6, page 237]. Les données traitées sont reportées dans le tableau 1. Elles regroupent les notes obtenues par six étudiants en mathématiques et en physique, lors de deux semestres consécutifs (M1, M2, P1, et P2). Certaines notes sont précisément connues, d'autres sont seulement connues sous forme d'intervalle, ou sous forme linguistique, ou encore absentes.

	M1	M2	P1	P2
Tom	15	fairly good	unknown	[14,16]
David	9	good	fairly good	10
Bob	6	[10,11]	[13,20]	good
Jane	fairly good	very good	19	[10,12]
Joe	very bad	fairly bad	[10,14]	[14]
Jack	1	[4,6]	9	[6,9]

TAB. 1 – Jeu de données des étudiants : tableau de notes.

On choisit de représenter chaque note sous forme d'un nombre flou trapezoïdal. Les appréciations linguistiques sont caractérisées par les fonctions d'appartenance représentées figure 2.
Un réseau comprenant deux cellules cachées ($q = 2$) a permis d'obtenir une représentation plane des données. L'examen de la matrice de poids (tableau 2) permet une interprétation des axes : le premier axe est lié aux performances en mathématiques alors que le second est plutot lié à la physique.
Sur le plan de projection représenté figure 3, on retouve de nombreuses caractéristiques importantes contenues dans les données originales. Par exemple, Jack, avec des notes dans l'ensemble assez mauvaises et précises, est bien localisé dans la partie basse de la figure. La grande dispersion de Tom le long du second

	axis 1	axis 2
M1	0.80	-0.07
M2	0.57	0.17
P1	0.00	0.95
P2	0.08	0.13

TAB. 2 – Valeurs des poids B_{ij} pour le jeu de données des étudiants.

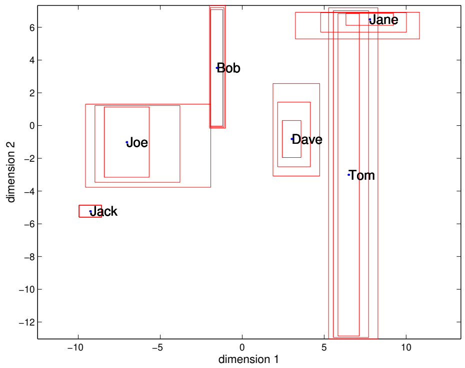

FIG. 3 – Représentation des étudiants dans le plan (supports, noyaux, et coupes de niveau 0.5).

axe s'explique par le fait qu'une de ses notes en physique est inconnue. Bob est très précisément situé sur l'axe lié aux mathématiques et de façon moins précise sur le second.
Une comparaison des données d'entrée et des données reconstruites est donnée figure 4. Elle revèle quels sont les aspects qui sont bien préservés dans la représentation compressé et ceux qui ne le sont pas : par exemple, la note de Bob dans la matière P1 est bien reconstruite alors que celle de Jack dans la matière P2 ne l'est pas. Les erreurs totales de reconstruction pour les six étudiants sont reportés table 3. Elles renseignent sur la confiance que l'on peut avoir dans la représentation des différents étudiants.

Tom	David	Bob	Jane	Joe	Jack
20.97	13.00	10.93	10.66	7.65	14.22

TAB. 3 – Erreurs de reconstruction pour le jeu de données des étudiants

4.2 Données sensorielles

La méthode proposée a été appliquée à des données d'évaluation sensorielle dans le cadre d'un projet mené en collaboration avec PSA Peugeot-Citröen. Les objets étudiés sont des sons enregistrés à l'intérieur de véhicules. Les données consistent en des scores at-

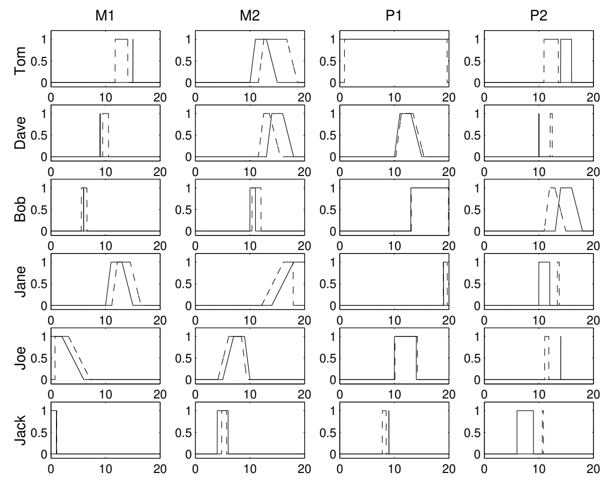

FIG. 4 – Jeu de données des étudiants. Données d'entrée (traits pleins) et données reconstruites (pointillés).

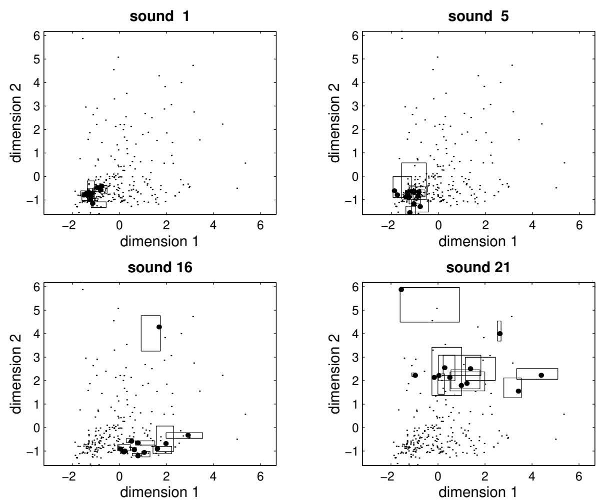

FIG. 5 – Jeu de données sensorielles. Chaque figure montre comment un son est perçu par les douze juges. Les cercles et les traits pleins représentent respectivement le noyau et l'alpha-coupe de niveau 0.5 des projections. Les points representent le centre de gravité des projections de toutes les autres paires (son,sujet) qui ne concernent pas le son considéré.

tribués par un panel de 12 juges décrivant la perception de 21 sons suivant 5 descripteurs. Chaque son a été présenté 3 fois à chaque sujet de telle sorte que l'on dispose d'une matrice à quatre entrées : sons × descripteurs × sujets × répétitions. Le but de l'étude était d'étudier la variabilité des réponses entre les différents panelistes et la répétabilité de chaque sujet au cours des répétitions. Pour cela, chacun des 21 × 12 couples (son, sujet) a été considéré comme un objet dans l'analyse. Pour chaque descripteur, les trois valeurs issues des répétitions ont servi à la construction de nombres flous triangulaires (qui est un cas particulier de nombre flou trapézoïdal) définis par une valeur minimale, maximale et centrale. L'ensemble de données initial consiste donc en 21 × 12 vecteurs composés de 5 nombres flous triangulaires. Une représentation en deux dimensions a été obtenue grâce à un réseau comprenant deux cellules cachées. Les poids, reportés tableau 4, montrent que les axes 1 et 2 sont fortement liés, respectivement, aux descripteurs 2 et 4. Une première partie des résultats est présentée sur la figure 5. Pour plus de clarté, les réponses des 12 sujets pour 4 sons différents sont présentées sur 4 figures différentes. Les quatre sons ont été choisis car ils permettent de mettre en évidence quatre comportement différents du panel. Le premier son est perçu de manière similaire par les juges et avec une variabilité faible au cours des répétitions. Les jugements pour le son 5 sont en assez bon accord mais avec une variabilité plus importante. Le son 16 présente les mêmes caractéristiques, mis à part un juge, qui est clairement en désaccord avec l'ensemble du groupe. Enfin, le son 21 semble être difficile à noter avec une variabilité inter et intra-juges importante. Une autre façon de présenter les résultats consiste à montrer l'intégralité des réponses fournies par un même juge. Sur la figure 6 sont présentés deux juges aux comportements extrêmes : le juge 4 utilise une échelle de notation de faible amplitude avec une bonne répétabilité. Le juge 12 au contraire utilise une plage de notation beaucoup large mais apparaît très variable dans ses évaluations. En étudiant attentivement toutes les représentations, il est donc possible de répondre à un certain nombre de questions qui se posent en analyse sensorielle : est-ce que les produits sont correctement distingués les uns des autres, quels sont les juges répétables, l'échelle de notation est-elle utilisée de façon pertinente, etc.

	axis 1	axis 2
x_1	-0.17	-0.16
x_2	0.92	0.00
x_3	0.32	-0.17
x_4	0.01	0.97
x_5	-0.13	-0.01

TAB. 4 – Jeu de données sensorielles. Poids B_{ij}.

5 Conclusion

Les données floues apparaissent naturellement dans un certain nombre de situations pour lesquelles

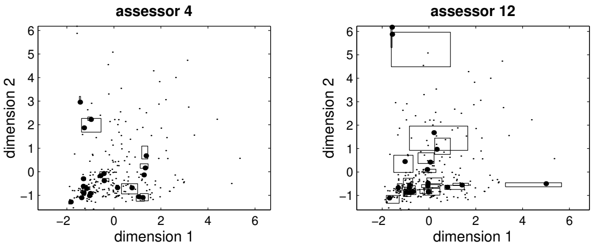

FIG. 6 – Jeu de données sensorielles. Chaque figure montre comment un juge particulier perçoit les 21 sons. Les cercles et les traits pleins représentent respectivement le noyau et l'alpha-coupe de niveau 0.5 des projections. Les points representent le centre de gravité des projections de toutes les autres paires (son,sujet) qui ne concernent pas le juge considéré.

l'incertitude ou l'imprécision des observations ne peuvent être ignorée. Par exemple, dans l'application à l'évaluation sensorielle décrite dans cet article, l'écart des évaluations entre répétitions est aussi important que la valeur centrale. De nouvelles méthodes d'analyse exploratoire de ce type de données doivent être développées pour prendre en compte cette complexité supplémentaire. La technique présentée ici est une extension de l'ACP classique. Elle exploite des développements récents concernant la capacité d'un réseau autoassociatif à realiser une compression de l'information exactement comme l'ACP, mais sans diagonalisation explicite de la matrice de variance-covariance. Les expériences menées tant sur données simulées que sur données réelles ont démontré la capacité de la méthode à fournir des représentation concises pertinentes de données multidimensionnelles complexes.

Remerciements

Nous remercions la société PSA Peugeot-Citröen pour avoir soutenu cette étude et fourni les données d'évaluation sensorielles.

Références

[1] P. Baldi and K. Hornik. Neural networks and principal component analysis : Learning from examples without local minima. *Neural Networks*, 2 :53–58, 1989.

[2] P. Baldi and K. Hornik. Learning in linear neural networks : A survey. *IEEE Transactions on Neural Networks*, 6(4) :837–858, 1995.

[3] J. C. Bezdek, J. Keller, R. Krishnapuram, and N. R. Pal. *Fuzzy models and algorithms for pattern recognition and image processing.* Kluwer Academic Publishers, Boston, 1999.

[4] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59 :291–294, 1988.

[5] P. Diamond and H. Tanaka. Fuzzy regression analysis. In R. Slowinski, editor, *Fuzzy sets in decision analysis, operations research and statistics*, pages 349–387. Kluwer Academic Publishers, Boston, 1998.

[6] D. Dubois and H. Prade. *Possibility Theory : An approach to computerized processing of uncertainty.* Plenum Press, New-York, 1988.

[7] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Boston, 1990.

[8] J. Gebhardt, M. A. Gil, and R. Kruse. Fuzzy set-theoretic methods in statistics. In R. Slowinski, editor, *Fuzzy sets in decision analysis, operations research and statistics*, pages 311–347. Kluwer Academic Publishers, Boston, 1998.

[9] R. J. Hathaway, J. C. Bezdek, and W. Pedrycz. A parametric model for fusing heterogeneous fuzzy data. *IEEE Transactions on Fuzzy Systems*, 4(3) :270–281, 1996.

[10] S. Heilpern. Representation and application of fuzzy numbers. *Fuzzy sets and systems*, 91 :259–268, 1997.

[11] W. Pedrycz, J. C. Bezdek, R. J. Hathaway, and G. W. Rogers. Two nonparametric models for fusing heterogeneous fuzzy data. *IEEE Transactions on Fuzzy Systems*, 6(3) :411–425, 1998.

Annexe

	Gradient de $\frac{\partial z_k^{(t)}}{\partial B_{ij}}$ $t \in \{1, \ldots, 4\}$			
	$i \neq k$		$i = k$	
	$B_{ij}B_{kj} > 0$	$B_{ij}B_{kj} < 0$	$B_{kj} > 0$	$B_{kj} < 0$
$\frac{\partial z_k^{(1)}}{\partial B_{ij}}$	$B_{kj} x_i^{(1)}$	$B_{kj} x_i^{(4)}$	$y_j^{(1)} + B_{kj} x_k^{(1)}$	$y_j^{(4)} + B_{kj} x_k^{(1)}$
$\frac{\partial z_k^{(2)}}{\partial B_{ij}}$	$B_{kj} x_i^{(2)}$	$B_{kj} x_i^{(3)}$	$y_j^{(2)} + B_{kj} x_k^{(2)}$	$y_j^{(3)} + B_{kj} x_k^{(2)}$
$\frac{\partial z_k^{(3)}}{\partial B_{ij}}$	$B_{kj} x_i^{(3)}$	$B_{kj} x_i^{(2)}$	$y_j^{(3)} + B_{kj} x_k^{(3)}$	$y_j^{(2)} + B_{kj} x_k^{(3)}$
$\frac{\partial z_k^{(4)}}{\partial B_{ij}}$	$B_{kj} x_i^{(4)}$	$B_{kj} x_i^{(1)}$	$y_j^{(4)} + B_{kj} x_k^{(4)}$	$y_j^{(2)} + B_{kj} x_k^{(4)}$

TAB. 5 – Dérivées partielles des sorties par rapport aux poids du réseaux.