Université de Montréal

Généralisation d'algorithmes de réduction de dimension

par
Jean-François Paiement

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maîtrise ès sciences (M.Sc.)
en informatique

Novembre, 2003

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé:

Généralisation d'algorithmes de réduction de dimension

présenté par:

Jean-François Paiement

a été évalué par un jury composé des personnes suivantes:

Patrice Marcotte
(président-rapporteur)

Yoshua Bengio
(directeur de recherche)

Balázs Kégl
(membre du jury)

Mémoire accepté le:

17 décembre 2003

Résumé

Mots clés : Statistiques, apprentissage statistique, algorithmes, apprentissage non supervisé, noyaux, réduction de dimension, forage de données.

On présente tout d'abord la notion de variété comme région de faible dimension contenant des observations situées dans un espace de haute dimension. Cette définition justifie l'élaboration d'algorithmes permettant d'exprimer les données dans un système de coordonnées de dimension égale à celle de la variété sur laquelle les données sont approximativement situées.

La notion de noyau comme mesure de similarité est par la suite formalisée. On constate que l'application d'un noyau à deux observations correspond à l'évaluation d'un produit scalaire dans un espace de Hilbert appelé espace de caractéristiques.

Une méthode de réduction de dimension linéaire est exposée ainsi que ses limites. Des algorithmes non linéaires de réduction de dimension et de segmentation permettent de s'affranchir de ces limites. Ces derniers ne fournissent cependant pas d'extension directe à des points hors échantillon.

L'étape fondamentale au sein des algorithmes présentés est la solution d'un système de vecteurs propres d'une matrice symétrique créée à partir d'un noyau dépendant des données. On conçoit ce problème comme le fait de trouver les fonctions propres d'un opérateur linéaire défini à partir du même noyau. On utilise alors la formule de Nyström, présente dans l'analyse en composantes principales à noyaux, afin de réduire la dimension des points hors échantillon sur la base des plongements obtenus à l'aide des algorithmes déjà mentionnés.

La qualité de la projection générée est comparée à la perturbation intrinsèque des algorithmes si on substitue certaines observations par d'autres tirées de la même distribution.

Summary

Keywords : Statistics, machine learning, algorithms, unsupervised learning, kernel methods, dimensionality reduction, data-mining.

First, we formally define manifold as a low dimensional region containing samples embedded in a high dimensional observation space. This definition justifies the conception of algorithms that allow expressing data in a coordinate system of dimension equal to the dimension of the manifold on which the data lies approximatively.

The notion of kernel as a measure of similarity is formalized. We see that the application of a kernel to a pair of observations corresponds to evaluate a scalar product in a Hilbert space, called the feature space.

Principal component analysis is described as a linear method for dimensionality reduction. Kernel principal component analysis allows to overcome the limitations of linear methods while offering a simple method that generalizes to out of sample observations. Isomap, LLE and spectral clustering can also generate low dimensional embeddings but don't provide straightforward extensions to out of sample observations.

The fondamental step in all these algorithms is to find the eigenvectors of a symmetric matrix generated by a data dependent kernel. We see this problem as finding the eigenfunctions of a linear operator defined by the same kernel. We use Nyström formula, which appears in kernel principal component analysis, to reduce out of sample points dimension on the basis of the embeddings already obtained by the algorithms. The projection quality is compared to the

intrinsic perturbation of the algorithms if we substitute some samples by ones drawn from the same distribution.

À mes parents
et à mon frère, homme de qualité

Remerciements

Avant tout, j'aimerais remercier mon directeur de recherche Yoshua Bengio, modèle pour moi de créativité et de constance indéfectible. Merci aussi à tous les membres du LISA, qui m'ont permis de travailler dans un environnement serein et stimulant. En particulier, j'aimerais remercier Yves Grandvalet pour avoir su répondre à toutes mes questions avec le sourire, Nicolas Chapados pour la qualité de la mise en page de ce mémoire, Christian Jauvin pour l'autre vision et Julien Keable pour la compagnie nocturne toujours sympathique.

Je tiens à souligner le soutien financier du FCAR et de PRECARN, des organismes grâce auxquels la poursuite d'études supérieures est beaucoup plus facile à envisager.

Merci Catherine, pour avoir enduré mon art vocal. Finalement, un merci profond à mes parents, qui m'ont encouragé sans hésitation dans toutes mes entreprises scientifiques et artistiques.

Table des matières

Liste des figures

Liste des tableaux

CHAPITRE 1

Introduction

Les informations visuelles perçues par un humain à un moment précis sont transmises au cerveau par plus de 10^6 nerfs optiques. Ces données peuvent être représentées par des vecteurs de très haute dimension. Cependant, le nombre de caractéristiques considérées par un individu pour évaluer visuellement la similitude entre deux objets est considérablement inférieur à cette dimension. Les structures cohérentes du monde induisent en effet de très fortes dépendances entre les différents stimuli traités par le cerveau. Le nombre de variables indépendantes décrivant l'information visuelle perçue est donc relativement faible par rapport au nombre d'informations distinctes transmises au cerveau a priori.

D'un point de vue plus pratique, les scientifiques sont régulièrement confrontés à des données en très haute dimension, qu'il s'agisse par exemple de patrons climatiques globaux, de spectres stellaires ou de distributions de gènes humains. Les chercheurs doivent donc tâcher de trouver des structures pertinentes de plus faible dimension cachées au sein des observations de haute dimension dont ils disposent (LITTMAN, SWAYNE, DEAN et BUJA 1992).

Lorsque certaines variables observées sont dépendantes, la dimension n de l'espace de données (ou nombre de variables observées) peut être supérieure au

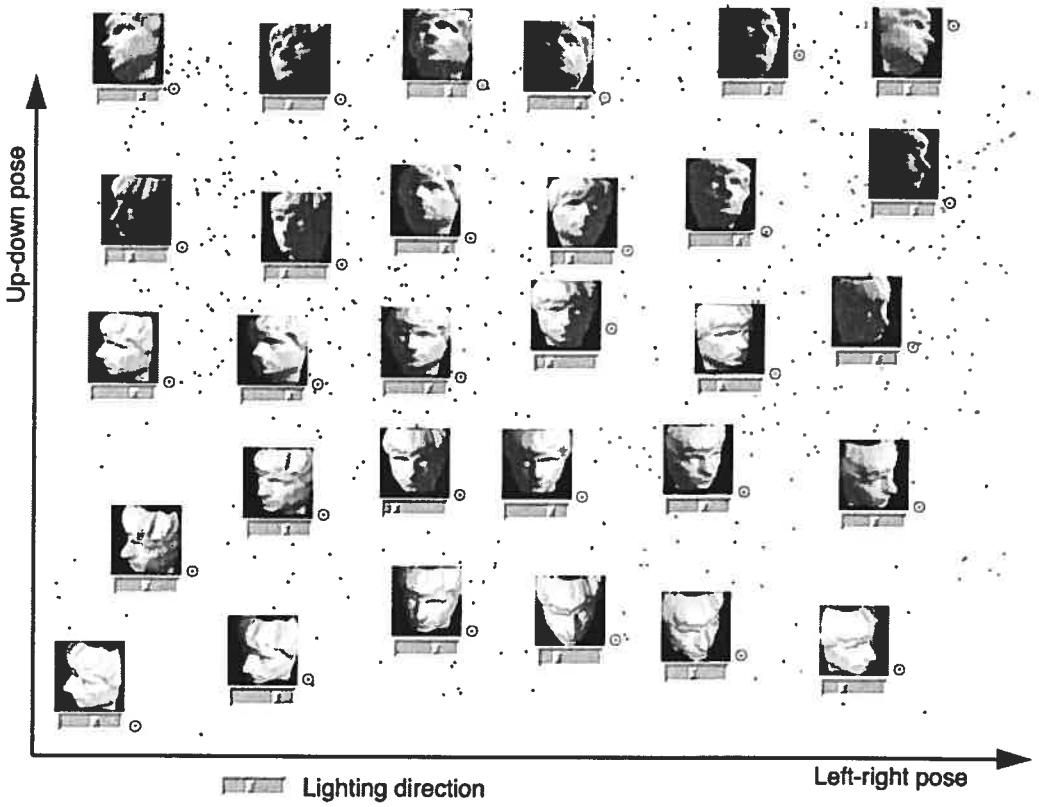

Figure 1.1 – *L'algorithme Isomap (présenté à la section 4.2) projette des images représentées par des vecteurs de dimension 4096 (correspondant au nombre de pixels d'une image) vers un espace de dimension 3 en ne considérant qu'un nombre fini d'images.*

nombre de variables indépendantes d réellement nécessaires pour représenter les données. Dans un contexte statistique, n est appelé la *dimension superficielle* (ou dimension d'observation) des données et d est plutôt défini comme étant la *dimension intrinsèque* des données. La quantité d peut être considérée comme le nombre de degrés de liberté du phénomène représenté par les données.

Supposons qu'une caméra capte plusieurs photos d'un même visage avec différents angles d'élévation, de rotation et d'éclairage, comme à la figure 1.1, empruntée à (TENENBAUM, SILVA et LANGFORD 2000). Supposons aussi que

ces images soient représentées en mémoire par 4096 pixels. Il est évident que cette représentation n'est pas la plus compacte possible et qu'une représentation en 3 dimensions telle qu'illustrée est optimale si on connaît la façon de reconstruire une image à partir de ses coordonnées dans l'espace de dimension réduite.

L'algorithme de réduction de dimension idéal parviendrait donc à trouver une projection des points de l'espace original en 4096 dimensions vers un espace à 3 dimensions où les relations entre les images seraient préservées et partir duquel il serait possible de reconstruire les images originales. Cette projection doit être estimée en ne considérant qu'un nombre fini d'images et en ne connaissant pas à l'avance leurs coordonnées dans l'espace d'arrivée. Il s'agit par conséquent d'un problème particulier du domaine informatique de l'*apprentissage non supervisé* (HINTON et SEJNOWSKI 1999), qui constitue l'étude des méthodes automatiques pouvant découvrir des structures cachées à partir des régularités statistiques de grands ensembles de données.

Des représentations où chaque dimension est significative peuvent en général être traitées beaucoup plus facilement que les observations originales par des algorithmes de classification ou de régression. Cependant, l'attrait le plus fondamental d'une telle représentation découle de l'indépendance statistique accrue entre les variables décrivant un échantillon. Scientifiquement, il est souvent beaucoup plus facile d'interpréter des données où chaque composante est essentielle à la caractérisation de chaque observation. Par conséquent, on peut espérer qu'une telle représentation exhibera les caractéristiques fondamentales d'une serie d'observations, ce qui constitue en fait un des objectifs fondamentaux de la recherche scientifique en général.

1.1 Quelques notions de topologie

Afin de rendre plus claire la distinction entre la dimension superficielle des données et leur dimension intrinsèque, il est utile d'introduire quelques définitions élémentaires de topologie.

Intuitivement, la topologie est l'étude mathématique des propriétés des objets préservées malgré les déformations, les torsions ou les étirements. Plus formellement, la topologie est l'étude des *espaces topologiques* (BISHOP et GOLDBERG 1980).

Définition 1.1 *Soit un ensemble X et un ensemble $T \subset 2^X$. X et T forment une* topologie *si les éléments de T satisfont*

1. $X \in T$ *et* $\emptyset \in T$;
2. $A, B \in T \implies A \cap B \in T$;
3. *L'union d'une collection d'ensembles de T est dans T.*

où l'ensemble 2^X est l'ensemble de tous les sous-ensembles de X.

Un ensemble X formant une topologie avec l'ensemble T est appelé un espace topologique (MUNKRES 1975). Un élément de T est appelé *sous-ensemble ouvert* de X. Le mot ouvert est souvent utilisé au lieu de sous-ensemble ouvert.

Définition 1.2 *Soit X un espace topologique, $N \in 2^X$ et $x \in X$. On dira que N est un* voisinage de x *s'il existe un ouvert U tel que $x \in U$ et $U \subset N$.*

Définition 1.3 *Une* boule ouverte *de centre $\boldsymbol{x} \in \mathbb{R}^n$ et de rayon $r > 0$ est l'ensemble $\mathbb{B}(\boldsymbol{x}, r)$ des points $y \in \mathbb{R}^n$ tels que $||\boldsymbol{x} - \boldsymbol{y}||_n < r$.*

Une boule ouverte dans R^n est un cas particulier du concept de voisinage dans un espace topologique. On dit aussi qu'une propriété P est vérifiée *localement* par un espace topologique X s'il existe pour chaque point de X un voisinage de ce point pour lequel P est vérifiée.

Il est possible de définir une relation d'équivalence entre espaces topologiques.

Définition 1.4 *Soit X et Y des espaces topologiques et soit $f : X \to Y$. f est un* homéomorphisme *si et seulement si*

1. *f est une bijection ;*
2. *f et f^{-1} sont continues.*

Le concept fondamental de *variété* peut finalement être introduit.

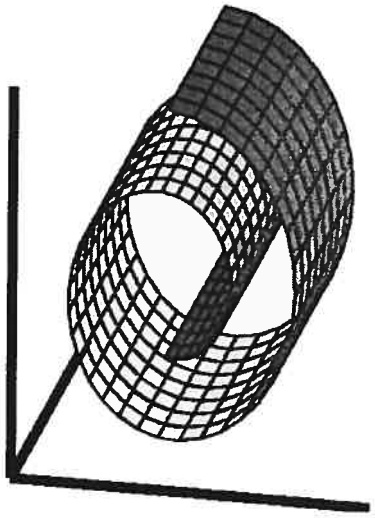

Figure 1.2 – *La surface représentée constitue une variété de dimension 2 plongée dans* $\mathbb{R}^3$.

Définition 1.5 *Une* variété V *de dimension* n *est un espace topologique localement euclidien, c'est-à-dire que pour tout* $x \in V$, *il existe un voisinage de* X *et un homéomorphisme de ce voisinage vers une boule ouverte dans* $\mathbb{R}^n$.

Une variété peut être visualisée comme un ensemble où chaque point peut-être considéré localement comme faisant partie d'un espace euclidien de plus faible dimension.

La figure 1.2 constitue un exemple simple de variété. On voit qu'une variété n'est pas nécessairement un sous-espace linéaire, auquel cas la surface représentée devrait être un plan passant par l'origine.

Les variétés mentionnées à partir de maintenant seront implicitement considérées comme des sous-ensembles de $\mathbb{R}^n$.

1.2 Apprentissage de variétés

Une *géodésique* entre deux points est une courbe localement de longueur minimale. Sans définir formellement les *métriques riemanniennes* (CHAVEL

1993; LEE 1997), qui définissent la notion de distance dans un espace, on notera tout de même que la distance dans un espace peut être définie autrement que par la notion de distance euclidienne.

On considère par exemple la distance entre deux villes sur la terre. La distance en voiture entre deux villes de la même région est approximativement égale à la distance euclidienne entre ces deux villes. Cependant, la distance sur la surface de la terre entre Montréal et Lausanne est très différente de la distance euclidienne qui les sépare. Supposons que la surface de la terre soit une variété, cette variété en deux dimensions est située dans un espace en trois dimensions, jusqu'à preuve du contraire. Ainsi, la distance considérée doit être située *sur* la variété et on constate donc que la distance entre deux villes est en fait la longueur de la géodésique située sur cette variété.

Plus généralement, supposons que l'on dispose d'un ensemble fini de données de dimension n mais de dimension intrinsèque d beaucoup plus faible. Les observations sont par conséquent approximativement situées sur une variété V de dimension d avec $V \subset \mathbb{R}^n$. La figure 1.3 illustre le cas d'une variété en deux dimensions dans un espace euclidien en trois dimensions traitée par l'algorithme Isomap (TENENBAUM, SILVA et LANGFORD 2000).

Le chapitre 3 présente l'algorithme d'analyse en composantes principales, qui réalise une réduction de dimension en projetant les données dans un sous-espace linéaire de dimension inférieure à la dimension d'observation. L'utilisation de cet algorithme est légitime si les données observées sont approximativement situées sur une variété linéaire. L'algorithme de positionnement multidimensionel est présenté par la suite. Ce dernier réalise indirectement une réduction de dimension linéaire à partir de distances.

Les déficiences des méthodes linéaires ont entraîné l'élaboration d'algorithmes pouvant approximer les données par des variétés non linéaires. Les noyaux, présentés au chapitre 2, fournissent des mesures de similitude non linéaires pouvant aisément être incorporées à des algorithmes linéaires déjà existants. Les noyaux définissent implicitement des espaces de caractéristiques non linéaires des observations.

Au lieu de considérer les observations directement, les algorithmes mo-

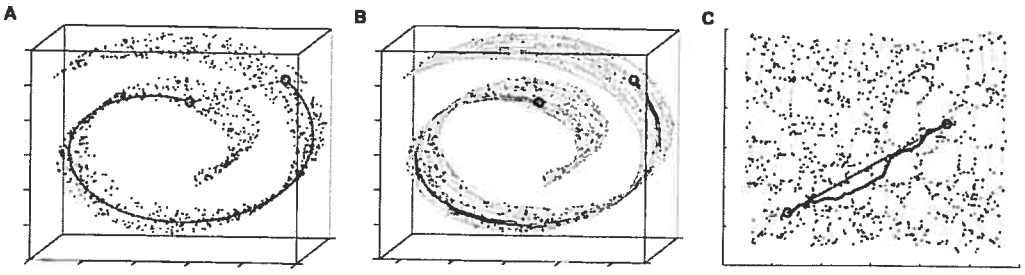

Figure 1.3 – *(A) Des données synthétiques sont tirées exactement d'une variété non linéaire en 2 dimensions dans un espace en 3 dimensions. La distance euclidienne (en pointillés) ne correspond pas nécessairement à la notion de distance souhaitée (trait plein.) (B) Dans Isomap, on définit plutôt la distance comme la longueur du plus court chemin dans un graphe reliant les plus proches voisins de chaque point. La géodésique entre deux points correspondant à cette distance est illustrée. (C) L'algorithme doit pouvoir projeter les points dans un espace de plus faible dimension tel que les distances euclidiennes dans cet espace deviennent approximativement égales aux longueurs des géodésiques dans l'espace original.*

difiés par l'utilisation d'un noyau évoluent dans un espace de caractéristiques non linéaires des observations. L'analyse en composantes principales à noyau, l'algorithme Isomap et le plongement localement linéaire sont présentés au chapitre 4. La segmentation spectrale est introduite par la suite.

Malheureusement, la plupart de ces algorithmes fournissent un positionnement en plus faible dimension pour les points d'un échantillon fixé au départ, mais n'offrent pas de méthode directe et non coûteuse de généralisation à de nouvelles observations. On constate cependant que les algorithmes non linéaires considérés sont en fait des cas particuliers d'un algorithme plus général, basé sur le calcul des fonctions propres d'un opérateur linéaire. Cette généralisation, présentée au chapitre 5, permet d'appliquer les algorithmes non linéaires de réduction de dimension déjà existants à de nouvelles données avec un coût linéaire en fonction du nombre de données considérées au départ. Cette innovation constitue la contribution majeure de ce mémoire.

CHAPITRE 2

Noyaux

Considérant l'incapacité pour les algorithmes linéaires de résoudre bon nombre de problèmes généraux, il semble judicieux de définir des mesures de similarité non linéaires entre les observations. Dans ce chapitre, on considère un espace d'observations $\mathcal{X}$ qui n'est pas nécessairement un sous-ensemble de $\mathbb{R}^n$. Cette approche peut éventuellement permettre de généraliser les algorithmes présentés à des situations où des représentations vectorielles des objets étudiés ne sont pas directement disponibles.

Considérons un *espace de caractéristiques* (feature space en anglais) $\mathcal{H}$ relié à l'espace $\mathcal{X}$ des observations par une application

$$\begin{aligned} \Phi : \quad & \mathcal{X} \to \mathcal{H} \\ & \boldsymbol{x} \mapsto \Phi(\boldsymbol{x}). \end{aligned}$$

L'espace de caractéristiques, de dimension arbitraire et même potentiellement infinie, peut être constitué de n'importe quelle combinaison possiblement non linéaire des composantes des observations.

Un *noyau* (kernel en anglais) est un mesure de similarité (ou de dissimila-

rité) de la forme

$$k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$(x, x') \mapsto k(x, x').$$

On montre que tout noyau répondant à certaines contraintes définit implicitement un espace de caractéristiques. Par la suite, on constate qu'il est possible d'utiliser la plupart des algorithmes linéaires connus dans un espace de caractéristiques plutôt que dans l'espace original des observations. On profite de ce chapitre pour introduire diverses notations et définitions nécessaires à la compréhension de la littérature sur les noyaux.

2.1 Espaces de Hilbert

Il est tout d'abord nécessaire d'introduire quelques notions d'analyse fontionnelle (REED et SIMON 1980; KOLMOGOROV et FOMIN 1961) qui combinent différents concepts d'analyse et d'algèbre linéaire.

On suppose les notions d'espace vectoriel, de norme et de produit scalaire connues (LANG 1989). Un *espace préhilbertien* est un espace vectoriel muni d'un produit scalaire. Un *espace normé* est un espace vectoriel muni d'une norme.

Définition 2.1 *Une suite* $(\boldsymbol{x}_i)_{i\in\mathbb{N}}$ *dans un espace normé* $\mathcal{H}$ *est appelée* suite de Cauchy *(*RUDIN *1995) si pour tout* $\epsilon > 0$, *il existe* $N \in \mathbb{N}$ *tel que pour tout* $n, m > N$, *on a* $||\boldsymbol{x}_n - \boldsymbol{x}_m|| < \epsilon$.

On dit qu'une suite de Cauchy converge *vers le point* $\boldsymbol{x} \in \mathcal{H}$ *si* $||\boldsymbol{x}_n - \boldsymbol{x}|| \to 0$ *quand* $n \to \infty$.

Définition 2.2 *Un espace* $\mathcal{H}$ *est* complet *si toutes les suites de Cauchy dans l'espace convergent.*

Un espace de Hilbert *est un espace préhilbertien complet.*

On dit qu'un *sous-ensemble dense* S d'un espace $\mathcal{H}$ est tel que chaque élément de $\mathcal{H}$ est la limite d'une suite dans S. Un espace de Hilbert est *séparable* s'il possède un sous-ensemble dense dénombrable.

2.2 Représentation des similarités

Traditionnellement, le problème d'associer un espace de caractéristiques à un noyau a été étudié en analyse fonctionnelle (BERG, CHRISTENSEN et RESSEL 1984; ARONSZAJN 1950; SAITOH 1988).

L'approche adoptée ici (SCHÖLKOPF et SMOLA 2002) se veut une introduction aux idées de base de l'étude des représentations des noyaux par des espaces de Hilbert.

Définition 2.3 *Soit une fonction* $k : \mathcal{X}^2 \to \mathbb{K}$ *(où* $\mathbb{K} = \mathbb{C}$ *ou* $\mathbb{K} = \mathbb{R}$*) et des observations* $\boldsymbol{x}_1, \ldots \boldsymbol{x}_p \in \mathcal{X}$*. La matrice* K *de taille* $p \times p$ *avec éléments*

$$K_{ij} := k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

est appelée matrice de Gram *de* k *en fonction de* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$.

où $\mathbb{C}$ est le corps des complexes et $\mathbb{R}$ le corps des réels.

Définition 2.4 *Une matrice complexe* K *de taille* $p \times p$ *satisfaisant*

$$\sum_{i,j} c_i \bar{c}_j K_{ij} \geq 0 \tag{2.1}$$

for tout $c_i \in \mathbb{C}$ *est dite* semi définie positive. *De même, une matrice réelle* K *de taille* $p \times p$ *satisfaisant (2.1) pour tout* $c_i \in \mathbb{R}$ *est dite* semi définie positive *(on écrit aussi* $K \geq 0$*).*

Une matrice symétrique est semi définie positive si et seulement si toutes ses valeurs propres ne sont pas négatives (LANG 1989).

Définition 2.5 *Soit* $\mathcal{X}$ *un ensemble non vide. Une fonction* k *de domaine* $\mathcal{X} \times \mathcal{X}$ *pour laquelle* $\boldsymbol{x}_i, \ldots, \boldsymbol{x}_p$ *induit une matrice de Gram définie positive pour tout* p *et tout* $\boldsymbol{x}_i, \ldots, \boldsymbol{x}_p \in \mathcal{X}^p$ *est appelée* noyau semi défini positif.

Il arrive souvent dans la littérature que cette caractéristique d'un noyau ne soit pas mentionnée explicitement et qu'on utilise abusivement le mot noyau en référant à un noyau semi défini positif.

Pour tout noyau semi défini positif,

$$k(x,x) \geq 0 \; \forall x \in \mathcal{X}$$

et

$$k(x_i,x_j) = \overline{k(x_j,x_i)}.$$

2.2.1 L'application des noyaux reproduisants

Soit k un noyau semi défini positif réel et $\mathcal{X}$ un ensemble non vide. On pose

$$\mathbb{R}^{\mathcal{X}} := \{f : \mathcal{X} \to \mathbb{R}\}$$

et on définit l'application

$$\begin{aligned} \Phi : \; & \mathcal{X} \to \mathbb{R}^{\mathcal{X}} \\ & x \mapsto k(\cdot, x). \end{aligned} \tag{2.2}$$

Cette application associe donc une fonction à chaque élément de $\mathcal{X}$. Un élément est maintenant représenté par sa similitude avec tous les autres patrons de $\mathcal{X}$. Il est possible de construire un espace de caractéristiques à partir de Φ.

On construit un espace préhilbertien contenant $\Phi(\mathcal{X})$. On crée tout d'abord un espace vectoriel en prenant les combinaisons linéaires de la forme

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \tag{2.3}$$

où $m \in \mathbb{N}$, $\alpha_i \in \mathbb{R}$ et $x_1, \ldots, x_m \in \mathcal{X}$ sont arbitraires.

On définit le produit scalaire entre f et une autre fonction

$$g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j)$$

où $m' \in \mathbb{N}$, $\beta_j \in \mathbb{R}$ et $x'_1, \ldots, x'_m \in \mathcal{X}$ par

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j). \tag{2.4}$$

On peut montrer que ce produit scalaire est bien bilinéaire, symétrique et défini positif. On a aussi que $\langle \cdot, \cdot \rangle$ est un noyau semi défini positif dans $\mathbb{R}^{\mathcal{X}}$ (SCHÖLKOPF et SMOLA 2002).

Pour toutes les fonctions (2.3), on a

$$\langle k(\cdot, x), f \rangle = f(x). \tag{2.5}$$

En particulier,

$$\langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x'). \tag{2.6}$$

À cause de ces propriétés, les noyaux définis positifs sont aussi appelés *noyaux reproduisants.*

En combinant (2.2) et (2.6), on voit que

$$\langle \Phi(x), \Phi(x') \rangle = k(x, x').$$

Ainsi, l'espace préhilbertien $\mathcal{H}$ construit de cette façon peut être considéré comme un espace de caractéristiques associé à un noyau. L'application du noyau à deux éléments de $\mathcal{X}$ revient à calculer leur produit scalaire dans l'espace de caractéristiques $\mathcal{H}$.

Pour tout algorithme faisant intervenir un noyau semi défini positif k, on peut créer un algorithme alternatif en remplaçant k par un autre noyau semi défini positif $\tilde{k}$. Le cas le plus courant d'application de cette technique dans la littérature est celui où le noyau semi défini positif original k est en fait le produit scalaire usuel dans l'espace d'observation. L'analyse en composantes principales à noyau présentée à la section 4.1 constitue un exemple éloquent d'application fructueuse de cette technique.

2.2.2 Espaces de Hilbert des noyaux reproduisants

Il est possible de calculer des projections dans les espaces de Hilbert au même titre que dans les espaces vectoriels finis, tel que le stipule le théorème suivant (KOLMOGOROV et FOMIN 1961), présenté sans preuve.

Théorème 2.1 *Soit $\mathcal{H}$ un espace de Hilbert et M un sous-espace fermé. Tout $\boldsymbol{x} \in \mathcal{H}$ peut être caractérisé par $\boldsymbol{x} = \mathbf{z} + \mathbf{z}^{\perp}$, où $\mathbf{z} \in M$ et $\langle \mathbf{z}^{\perp}, \mathbf{t} \rangle = 0$ pour tout $\mathbf{t} \in M$. Le vecteur $\mathbf{z}$ est l'élément unique minimisant $||\boldsymbol{x} - \mathbf{z}||$ et est appelé la* projection $P\boldsymbol{x} := \mathbf{z}$ *de $\boldsymbol{x}$ sur M. L'opérateur de projection P est une application linéaire.*

On rappelle qu'un sous-ensemble M de $\mathcal{H}$ est *fermé* si le point d'accumulation de toute suite convergente de M est contenu dans M. Tout sous-ensemble fermé d'un espace de Hilbert est aussi un espace de Hilbert.

On peut donc souhaiter transformer l'espace préhilbertien construit à la section précédente en espace de Hilbert afin de pouvoir définir des projections dans l'espace de caractéristiques défini par un noyau donné. On considère l'espace préhilbertien de fonctions (2.3) muni du produit scalaire (2.4). Pour transformer cet espace en espace de Hilbert sur $\mathbb{R}$, il faut le compléter en considérant la norme correspondant au produit scalaire $||f|| := \sqrt{\langle f, f, \rangle}$. Il faut ajouter les points d'accumulation des suites qui convergent selon cette norme. À cause des propriétés (2.5) et (2.6), on appelle cet espace *espace de Hilbert des noyaux reproduisants* ou RKHS (*reproducing kernel Hilbert space*). On peut consulter (REED et SIMON 1980) pour une définition formelle des RKSH. On note toutefois qu'un RKSH détermine un noyau k de façon univoque.

2.2.3 Éléments de théorie des probabilités

Il est maintenant nécessaire d'exposer quelques notions de base de probabilités et mesure (BILLINGSLEY 1995; BREIMAN 1968; FELLER 1971).

Définition 2.6 *Soit $\mathcal{X}$ un ensemble non vide. Une collection $\mathcal{C}$ de sous-ensembles de $\mathcal{X}$ est appelée une* σ-algèbre *sur $\mathcal{X}$ si*

1. $\mathcal{X} \in \mathcal{C}$;
2. $C \in \mathcal{C}$ *implique que* $\bar{C} \in \mathcal{C}$;
3. $C_1, C_2, \cdots \in \mathcal{C}$ *implique que l'union dénombrable* $\bigcup_{i=1}^{\infty} C_i \in \mathcal{C}$.

où $\bar{C} = \mathcal{X} \setminus \mathcal{C}$.

Une *mesure de probabilité* peut alors être définie comme une fonction d'une σ-algèbre vers l'intervalle $[0, 1]$.

Définition 2.7 *Soit $\mathcal{C}$ une σ-algèbre sur le domaine $\mathcal{X}$. Une fonction*

$$P : \mathcal{C} \to [0, 1]$$

est appelée mesure de probabilité *si elle est normalisée,*

$$P(\mathcal{X}) = 1,$$

et σ-additive, ce qui veut dire que pour des ensembles $C_1, C_2, \cdots \in \mathcal{C}$ mutuellement disjoints ($C_i \cap C_j = \emptyset$ si $i \neq j$), on a

$$P\left(\bigcup_{i=1}^{\infty} C_i\right) = \sum_{i=1}^{\infty} P(C_i).$$

Une mesure de probabilité sans la contrainte de normalisation est simplement appelée une *mesure.*

On dit qu'un couple $(\mathcal{X}, \mu)$ est un *espace à mesure finie* si $\mathcal{X}$ est un ensemble avec une σ-algèbre et μ est une mesure satisfaisant $\mu(\mathcal{X}) < \infty$. Ainsi, μ est une mesure de probabilité à un facteur près.

Soit $f : \mathbb{R} \to \mathbb{R}$. On dit que f est *mesurable* si pour tout intervalle $[a, b] \subset \mathbb{R}$, $f^{-1}([a, b]) \in \mathcal{C}$. L'extension aux fonctions à valeurs vectorielles se fait composante par composante.

On rappelle que la fonction p est appelée la *densité* de la mesure de probabilité P si pour tout $C \in \mathcal{C}$, on a

$$P(C) = \int_C p(x)dx.$$

On peut finalement introduire la notion d'intégrale par rapport à une mesure. Soit $f : \mathbb{R}^n \to \mathbb{R}$ mesurable. On note

$$\int_C f(x)dP(x) \tag{2.7}$$

l'intégrale d'une fonction par rapport à la mesure P.

Dans le cas où p est la densité de P, (2.7) est égal à

$$\int_C f(x)p(x)dx.$$

2.2.4 L'application des noyaux de Mercer

Un espace normé l_p^N est un espace vectoriel identique à $\mathbb{R}^N$, mais muni d'une norme-p définie par

$$||x||_{l_p^N} := ||x||_p = \left(\sum_{j=1}^{N} |x_j|^p\right)^{1/p}.$$

Pour $p = \infty$, on a

$$||x||_{l_\infty^N} := ||x||_\infty = \max_{j=1,\ldots,N} |x_j|.$$

On note l_p le cas où $N \to \infty$ et on ne considère que les séquences de norme-p finie. Le maximum est alors remplacé par un supremum.

Étant donné un ensemble $\mathcal{X}$ avec une σ-algèbre, une mesure μ sur $\mathcal{X}$, un scalaire p tel que $1 \leq p \leq \infty$ et une fonction $f : \mathcal{X} \to \mathbb{R}$, on définit

$$||f||_{L_p(\mathcal{X})} := ||f||_p := \left(\int |f(x)|^p d\mu(x)\right)^{1/p}$$

si l'intégrale existe et

$$||f||_{L_\infty(\mathcal{X})} := ||f||_\infty := \operatorname*{ess\,sup}_{x\in\mathcal{X}} |f(x)|$$

où ess sup est le supremum essentiel, c'est-à-dire la plus petite borne supérieure

de $|f(x)|$ presque partout[1].

On définit aussi

$$L_p(\mathcal{X}) := \{f : \mathcal{X} \to \mathbb{R} \mid ||f||_{L_p(\mathcal{X})<\infty}\}.$$

L'introduction fastidieuse de toute cette terminologie permet d'introduire le théorème de Mercer (MERCER 1909; KÖNIG 1986), qui définit un autre espace de caractéristiques associé à un noyau que celui présenté à la section 2.2.1. Ce théorème a joué un rôle crucial dans l'élaboration des machines à vecteur de support (VAPNIK 1998; CRISTIANINI et SHAWE-TAYLOR 2000). La définition de l'espace de caractéristiques associée à un noyau donnée par le théorème de Mercer constitue aussi la base des techniques de généralisation des algorithmes de réduction de dimension développées au chapitre 5.

Théorème 2.2 (Mercer) *Supposons que $k \in L_\infty(\mathcal{X}^2)$ est une fonction symétrique à valeurs réelles telle que*

$$\begin{aligned} T_k &: L_2(\mathcal{X}) \to L_2(\mathcal{X}) \\ (T_k f)(x) &:= \int_{\mathcal{X}} k(x,x')f(x')d\mu(x') \end{aligned}$$

est semi défini positif, c'est-à-dire que pour tout $f \in L_2(\mathcal{X})$, on a

$$\int_{\mathcal{X}^2} k(x,x')f(x)f(x')d\mu(x)d\mu(x') \geq 0.$$

Soient $\psi_j \in L_2(\mathcal{X})$ les fonctions propres de T_k solutions de

$$(T_k\psi_j)(x) = (\lambda_j\psi_j)(x)$$

normalisées et orthogonales associées aux valeurs propres $\lambda_j > 0$, triées en ordre décroissant.

On a alors que

$$(\lambda_j)_{j\in\mathbb{N}} \in l_1$$

[1] *Presque partout* ou *presque tout* signifie *excepté pour les sous-ensembles de mesure nulle* dans cette section.

et que

$$k(x,x') = \sum_{j=1}^{N_{\mathcal{H}}} \lambda_j \psi_j(x)\psi_j(x') \tag{2.8}$$

est valide pour presque tout (x,x'). *On a* $N_{\mathcal{H}} \in \mathbb{N}$ *ou* $N_{\mathcal{H}} = \infty$. *Dans le dernier cas, les séries (2.8) convergent absolument et uniformément pour presque tout* (x,x').

L'équation (2.8) indique que $k(x,x')$ correspond à un produit scalaire dans $l_2^{N_{\mathcal{H}}}$ étant donné que $k(x,x') = \langle \Phi(x), \Phi(x') \rangle$ avec

$$\begin{array}{rccl} \Phi: & \mathcal{X} & \to & l_2^{N_{\mathcal{H}}} \\ & x & \mapsto & (\sqrt{\lambda_j}\psi_j(x))_{j=1,\ldots,N_{\mathcal{H}}} \end{array}$$

pour presque tout $x \in \mathcal{X}$ (WAHBA 1990).

On montre dans (SCHÖLKOPF 1997) que la convergence uniforme des séries implique que pour tout $\epsilon > 0$, il existe $n \in \mathbb{N}$ tel que même si $N_{\mathcal{H}} = \infty$, k peut être approximé avec une précision de ϵ comme un produit scalaire dans $\mathbb{R}^n$. En effet, pour tout $\epsilon > 0$, il existe une application Φ_n telle que $|k(x,x') - \langle \Phi^n(x), \Phi^n(x') \rangle| < \epsilon$ pour presque tout $x, x' \in \mathcal{X}$, où $\Phi^n : x \mapsto (\sqrt{\lambda_1}\psi_1(x), \ldots, \sqrt{\lambda_n}\psi_n(x))$. Le produit scalaire peut donc être de dimension finie pour toute précision arbitraire.

En définitive, on constate qu'il existe plusieurs façons de définir l'espace de caractéristiques associé à un produit scalaire. Cependant, si Φ_1 et Φ_2 sont des applications dans les espaces de caractéristiques $\mathcal{H}_1$ et $\mathcal{H}_2$ associées au produit scalaire $k(x,x')$, on aura toujours

$$k(x,x') = \langle \Phi_1, \Phi_1 \rangle_{\mathcal{H}_1} = \langle \Phi_2, \Phi_2 \rangle_{\mathcal{H}_2}.$$

En pratique, il n'est donc pas nécessaire de définir explicitement un espace de caractéristiques associé à un noyau donné si on n'est intéressé qu'au produit scalaire dans cet espace.

CHAPITRE 3

Méthodes linéaires de réduction de dimension

L'*analyse en composantes principales* (ACP) (HOTELLING 1933) est une méthode de réduction de dimension linéaire classique consistant à projeter les échantillons sur les axes de variance maximale des données.

Le *positionnement multidimensionnel* (MDS) (COX et COX 1994; MARDIA, KENT et BIBBY 1979) construit plutôt une configuration de m points dans $\mathbb{R}^d$ à partir d'informations sur les distances entre m objets auquels ces points sont associés.

3.1 Analyse en composantes principales

L'objectif visé par la méthode ACP est d'identifier les dépendances linéaires sous-jacentes à une observation stochastique[1] multivariée afin d'en obtenir une description de plus faible dimension, tel qu'explicité à la section 1.2.

[1] Aléatoire.

3.1.1 Régression linéaire

Supposons un ensemble fini de points $(\boldsymbol{x_i}, \boldsymbol{y_i})$. (PEARSON 1901) a introduit la *régression linéaire* (ou méthode linéaire des moindres carrés) comme une méthode consistant à trouver des hyperplans définis par $f(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta x}$ en choisissant des paramètres minimisant l'*erreur quadratique de reconstruction* donnée par

$$\sum_i (\boldsymbol{y_i} - f(\boldsymbol{x_i}))^2. \tag{3.1}$$

Comme $f(x)$ est linéaire, la minimisation de (3.1) peut être réalisée en annulant ses dérivées partielles, donnant un système d'équations linéaires pouvant être résolu analytiquement. L'utilisation de cette technique afin d'analyser les structures de corrélation entre plusieurs variables aléatoires a été proposée dans (HOTELLING 1933).

3.1.2 Minimisation de l'erreur quadratique moyenne de reconstruction

La minimisation de l'erreur quadratique moyenne de reconstruction est équivalente à la maximisation de la variance de la projection dans l'espace de plus faible dimension. En effet, considérons un vecteur aléatoire $\boldsymbol{x} = (x_1, \ldots, x_n)$ de moyenne $E[\boldsymbol{x}] = 0$ et de matrice de covariance

$$C = E[\boldsymbol{x}\boldsymbol{x}^T] \in \mathbb{R}^{n \times n}. \tag{3.2}$$

ACP réalise une transformation linéaire orthogonale des données produite par

$$\boldsymbol{y} = W\boldsymbol{x}$$

où les colonnes orthonormales de W constituent la base d'un sous-espace $\mathcal{L} \subset \mathbb{R}^d$ et $WW^T = I$. La reconstruction de $\boldsymbol{x}$ à partir de $\boldsymbol{y}$ est donnée par

$$\hat{\boldsymbol{x}} = W^T\boldsymbol{y} = W^TW\boldsymbol{x}$$

Comme mentionné précédemment, ACP cherche à minimiser l'erreur quadratique moyenne de reconstruction

$$\begin{aligned} J_e &= E[||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2] \\ &= E[\mathrm{tr}\{(\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^T\}] \\ &= \mathrm{tr}(C) - \mathrm{tr}(WCW^T) \end{aligned} \tag{3.3}$$

où on a utilisé les identités $\mathrm{tr}(A) = \mathrm{tr}(A^T)$ et $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

On voit que le dernier terme dans 3.3 (que nous appellerons J_v) est égal à la variance des $\boldsymbol{y}$, qui est égale à la variance de la projection $\hat{x}$. En effet,

$$\begin{aligned} J_v &:= \mathrm{tr}(WCW^T) = E[\mathrm{tr}(\boldsymbol{y}\boldsymbol{y}^T)] = \textstyle\sum_{i=1}^{m} y_i^2 \\ &= \mathrm{tr}(W^TWCW^TW) = E[\mathrm{tr}(\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^T)] = \textstyle\sum_{i=1}^{n} \hat{x}_i^2. \end{aligned}$$

Comme stipulé précédemment, la maximisation de la variance de la projection est donc équivalente à la minimisation de l'erreur quadratique moyenne de reconstruction.

Le théorème suivant (démontré dans (DIAMANTARAS et KUNG 1996)) indique de quelle façon ACP parvient à optimiser ce critère.

Théorème 3.1 *Soit les valeurs propres λ_i de C telles que $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ et leurs vecteurs propres normalisés $\boldsymbol{e}_i$ correspondants. Sous la contrainte $WW^T = I$, l'erreur quadratique moyenne de reconstruction J_e est minimisée par une solution de la forme*

$$W_{opt} = T[\pm\boldsymbol{e}_1, \cdots, \pm\boldsymbol{e}_m]^T$$

où $T \in \mathbb{R}^{m \times m}$ est une matrice orthogonale quelconque.

Les vecteurs propres orthogonaux de C correspondant aux plus grandes valeurs propres (les lignes de W_{opt}) sont appelés les *vecteurs propres principaux*. Les composantes $y_1, \ldots, y_m$ du vecteur aléatoire $\boldsymbol{y}$ sont appelés *composantes principales* de $\boldsymbol{x}$. Ces variables aléatoires sont non corrélées car

$$E[y_i y_j] = \boldsymbol{e}_i^T C \boldsymbol{e}_j = 0.$$

Leur variance est donnée par

$$E[y_i^2] = \boldsymbol{e}_i^T C \boldsymbol{e}_i = \lambda_i.$$

Ainsi, supposons que nous disposions de données en dimension n et que nous voulions exprimer ces données en $d < n$ dimensions de façon à minimiser l'erreur quadratique moyenne de reconstruction. Il suffit alors de calculer la matrice C de covariance des points et de projeter chaque point sur les d vecteurs propres principaux de la matrice de covariance. Les coordonnées ainsi obtenues pour chaque point seront non corrélées et de variance maximale, comme illustré à la figure 3.1.

3.2 Positionnement multidimensionnel

Le positionnement multidimensionnel permet de construire une configuration de m points dans $\mathbb{R}^d$ à partir des distances entre m objets auquels ces points sont associés. Dans ce contexte, on n'observe plus les points directement dans l'espace d'origine mais plutôt les $\frac{1}{2}m(m-1)$ distances correspondant à ces points. Il est toujours possible de générer un positionnement de m points en m dimensions respectant exactement les distances fournies. MDS produit un positionnement des points dans $\mathbb{R}^d$, avec possiblement $d < m$, de façon à minimiser l'erreur quadratique moyenne entre les distances fournies à l'algorithme et celles des points générés.

Les distances fournies à MDS ne doivent pas nécessairement être euclidiennes et peuvent par conséquent représenter des dissimilitudes de tout genre entre des objets.

Toute translation, rotation ou réflexion d'une solution produite par MDS est une solution équivalente, car de telles transformations n'affectent pas les distances entre les points. En général, si les $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})$ représentent une solution en d dimensions donnée par MDS, alors

$$\boldsymbol{y}_i = T\boldsymbol{x}_i + \boldsymbol{b}$$

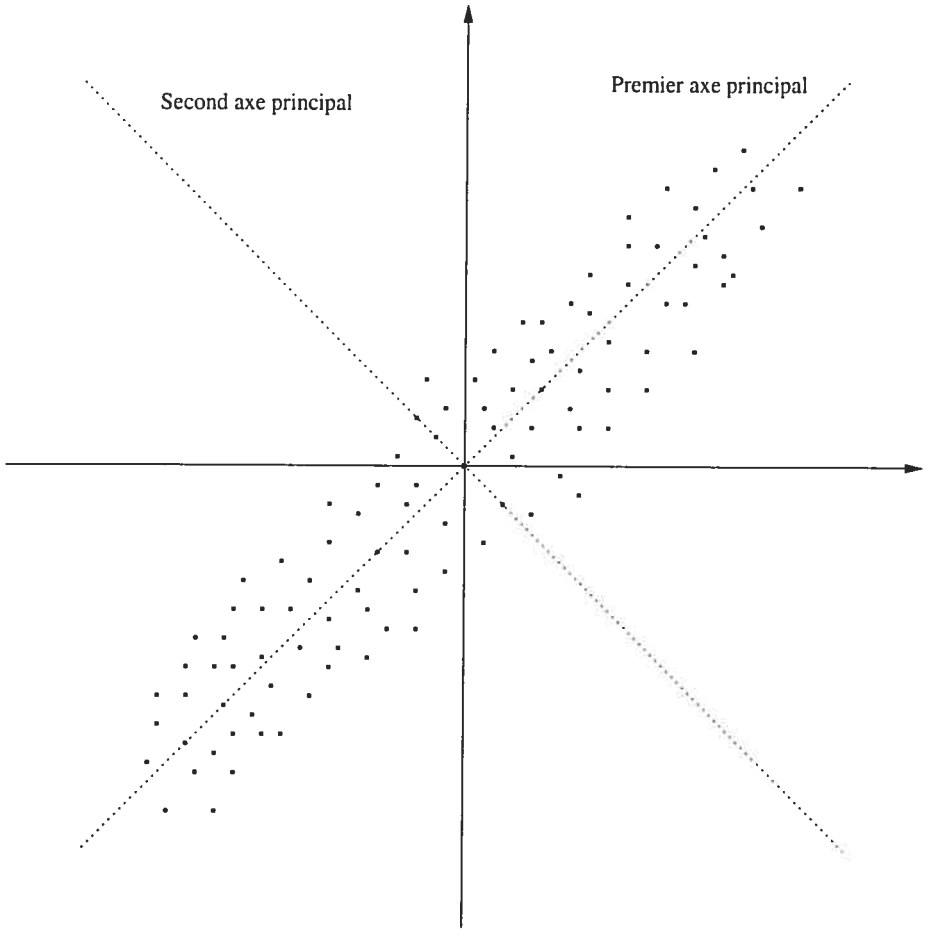

Figure 3.1 – *La projection des points sur l'axe de variance maximale (parallèle au vecteur propre principal de la matrice de covariance des données) minimise l'erreur quadratique de reconstruction. Intuitivement, la projection des points sur cet axe est la projection linéaire qui conserve le plus d'information sur les coordonnées originales des points.*

est aussi une solution, où T est orthogonale et $\boldsymbol{b}$ est un vecteur quelconque.

Définition 3.1 *(*Mardia, Kent *et* Bibby *1979) Une matrice* $D = (d_{ij}) \in \mathbb{R}^{m\times m}$ *est appelée* matrice de distance *si elle est symétrique, si* $d_{rr} = 0$ *et que* $d_{rs} \geq 0$ *pour* $r \neq s$. *De plus, une matrice de distance est* euclidienne *s'il existe une configuration de points dans un espace euclidien telle que les distances entre ces points sont données par* D, *c'est-à-dire qu'il existe un* d *et des points* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p \in \mathbb{R}^d$ *tels que*

$$d_{rs}^2 = (\boldsymbol{x}_r - \boldsymbol{x}_s)^T(\boldsymbol{x}_r - \boldsymbol{x}_s).$$

Pour toute matrice de distance $D = (d_{rs})$, on pose

$$A = (a_{rs}),\ a_{rs} = -\frac{1}{2}d_{rs}^2$$

et

$$M = HAH \tag{3.4}$$

où $H = I - n^{-1}\mathbf{1}\mathbf{1}^T$ est une matrice de centrage $m \times m$ et

$$\mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

Le théorème suivant, présenté dans (Mardia, Kent et Bibby 1979), permet de déterminer si une matrice D est euclidienne. Le cas échéant, il permet de trouver une configuration de points correspondante. Ce théorème est présenté avec preuve afin de mettre en lumière le rôle de la normalisation de D dans la transformation des distances en produits scalaires. Ce résultat a tout d'abord été démontré par (Schoenberg 1935) et (Young et Householder 1938). Son utilisation pour le positionnement multidimensionnel a été introduite par (Torgerson 1958) et les idées développées ont été considérablement amplifiées par (Gower 1966).

Théorème 3.2 *Soit D une matrice de distance et M définie par (3.4). D est euclidienne si et seulement si M est semi définie positive. En particulier, on a*

1. *Si D est la matrice des distances euclidiennes entre les points d'une configuration $Z = (\mathbf{z}_1, \ldots, \mathbf{z}_p)^T$, alors*

$$b_{rs} = (\mathbf{z}_r - \bar{\mathbf{z}})^T(\mathbf{z}_s - \bar{\mathbf{z}}),\ r, s = 1, \ldots, m. \tag{3.5}$$

   En forme matricielle, (3.5) devient $M = (HZ)(HZ)^T$, alors $M \geq 0$.

2. *Inversement, si M est semi définie positive de rang d, alors une configuration correspondant à M peut être construite. Soit $\lambda_1 > \cdots > \lambda_d$ les valeurs propres positives de M correspondant aux vecteurs propres $X = (\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(d)})$ normalisés par*

$$\boldsymbol{x}_{(i)}^T \boldsymbol{x}_{(i)} = \lambda_i,\ i = 1, \ldots, d.$$

   Alors les points P_r dans $\mathbb{R}^m$ avec coordonnées $\boldsymbol{x}_r = (x_{r1}, \ldots, x_{rm})^T$ sont séparés par les distances données par D.

Preuve On prouve tout d'abord 1. Supposons que

$$d_{rs}^2 = -2a_{rs} = (\mathbf{z}_r - \mathbf{z}_s)^T(\mathbf{z}_r - \mathbf{z}_s). \tag{3.6}$$

On peut écrire

$$M = HAH = A - d^{-1}AJ - d^{-1}JA + d^{-2}JAJ,$$

où $J = \mathbf{1}\mathbf{1}^T$. On a que

$$\tfrac{1}{d}AJ = \begin{bmatrix} \bar{a}_{1\cdot} & \ldots & \bar{a}_{1\cdot} \\ \vdots & & \vdots \\ \bar{a}_{d\cdot} & \ldots & \bar{a}_{d\cdot} \end{bmatrix},\ \tfrac{1}{d}JA = \begin{bmatrix} \bar{a}_{\cdot 1} & \ldots & \bar{a}_{\cdot d} \\ \vdots & & \vdots \\ \bar{a}_{\cdot 1} & \ldots & \bar{a}_{\cdot d} \end{bmatrix},\ \tfrac{1}{d^2}JAJ = \begin{bmatrix} \bar{a}_{\cdot\cdot} & \ldots & \bar{a}_{\cdot\cdot} \\ \vdots & & \vdots \\ \bar{a}_{\cdot\cdot} & \ldots & \bar{a}_{\cdot\cdot} \end{bmatrix}$$

où

$$\bar{a}_{r\cdot} = \frac{1}{n}\sum_{s=1}^{n} a_{rs}, \quad \bar{a}_{\cdot s} = \frac{1}{n}\sum_{r=1}^{n} a_{rs}, \quad \bar{a}_{\cdot\cdot} = \frac{1}{n^2}\sum_{r,s=1}^{n} a_{rs}. \tag{3.7}$$

Ainsi,

$$M_{rs} = a_{rs} - \bar{a}_{r\cdot} - \bar{a}_{\cdot s} + \bar{a}_{\cdot\cdot}$$

En utilisant (3.6) et (3.7), on a finalement que

$$M_{rs} = (\mathbf{z}_r - \bar{\mathbf{z}})^T(\mathbf{z}_s - \bar{\mathbf{z}}), \tag{3.8}$$

ce qui démontre 1. On dit que la génération de la matrice M constitue un *centrage double* de la matrice de distances D.

Pour démontrer 2, on suppose que $M \geq 0$ et on considère la configuration donnée par le théorème. Soit $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$ et soit $\Gamma = X\Lambda^{1/2}$. Les colonnes de Γ, $\gamma_{(i)} = \lambda_i^{-1/2}\boldsymbol{x}_{(i)}$ sont les vecteurs propres normalisés de M. Par le théorème de décomposition spectrale (LANG 1989),

$$M = \Gamma\Lambda\Gamma^T = XX^T.$$

Ainsi, $M_{rs} = \boldsymbol{x}_r^T\boldsymbol{x}_s$ et M est la matrice produit scalaire pour cette configuration. On montre que D est la matrice des distances entre les points de cette configuration. En utilisant (3.8), on a

$$\begin{aligned}(\boldsymbol{x}_r - \boldsymbol{x}_s)^T(\boldsymbol{x}_r - \boldsymbol{x}_s) &= \boldsymbol{x}_r^T\boldsymbol{x}_r - 2\boldsymbol{x}_r^T\boldsymbol{x}_s + \boldsymbol{x}_s^T\boldsymbol{x}_s \\ &= M_{rr} - 2M_{rs} + M_{ss} \\ &= a_{rr} - 2a_{rs} + a_{ss} \\ &= -2a_{rs} = d_{rs}^2\end{aligned}$$

car $a_{rr} = -\frac{1}{2}d_{rr}^2 = 0$ et $-2a_{rs} = d_{rs}^2$. ■

On peut aussi montrer que la configuration obtenue a un centre de gravité $\bar{\boldsymbol{x}} = 0$.

Ainsi, supposons qu'on dispose d'une matrice D et que cette matrice approxime les distances d'une configuration dans un espace euclidien de dimension d. Une configuration possible de dimension d est proposée par le

1. Construire la matrice $A = (-\frac{1}{2}d_{rs}^2)$ à partir de D ;
2. Construire la matrice M telle que $b_{rs} = a_{rs} - \bar{a}_{r\cdot} - \bar{a}_{\cdot s} + \bar{a}_{\cdot\cdot}$;
3. Trouver les d plus grandes valeurs propres $\lambda_1 > \cdots > \lambda_d$ de M et les vecteurs propres correspondants $X = (\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(d)})$ normalisés par $\boldsymbol{x}'_{(i)}\boldsymbol{x}_{(i)} = \lambda_i$;
4. Les coordonnées des points P_r sont données par $\boldsymbol{x}_r = (x_{r1}, \ldots, x_{rd})$, les lignes de X.

Tableau 3.1 – *Algorithme MDS classique.*

théorème 3.2. Il s'agit de choisir la configuration dans $\mathbb{R}^d$ telle que les coordonnées des points sont données par les d premiers vecteurs propres de M. Si les d premières valeurs propres de M sont grandes et positives et que les autres valeurs propres sont presques nulles, on peut espérer que les distances entre les points de cette configuration approximeront D de façon relativement précise. L'algorithme proposé est résumé au tableau 3.1.

Soit $\widehat{X}$ une configuration donnée par l'algorithme MDS et $\widehat{M}$ la matrice de produits scalaires, donnée par (3.4), calculée à partir des distances entre les points de cette configuration. On souligne dans (MARDIA 1978) qu'une mesure de dissimilitude entre M et $\widehat{M}$ est donnée par

$$\sum_{r,s=1}^{n} (b_{rs} - \hat{b}_{rs})^2 = \mathrm{tr}(M - \widehat{M})^2. \tag{3.9}$$

Il est montré dans (MARDIA, KENT et BIBBY 1979) que si D est une matrice de distance (pas nécessairement euclidienne), alors pour un d fixé, (3.9) est minimisée si $\widehat{M}$ est construite à partir de la configuration $\widehat{X}$ donnée par l'algorithme MDS explicité au tableau 3.1.

3.3 Déficiences des méthodes linéaires

L'ACP et MDS considèrent les distances euclidiennes dans l'espace d'observation. Si on suppose que les données ont une dimension intrinsèque m plus faible que la dimension d'observation n, l'ACP ne pourra trouver un système de coordonnées en m dimensions exact que si la variété à partir de laquelle sont tirées les données est en fait un *sous-espace linéaire*. Si les données sont plutôt tirées d'une variété non linéaire, comme à la figure 1.3 (**A**), l'ACP sera incapable d'exprimer les caractéristiques de cette variété et ne pourra pas par conséquent créer une représentation en plus faible dimension respectant les longueurs des géodésiques sur la variété.

Dans le même ordre d'idées, si les distances fournies à MDS sont les distances euclidiennes, cet algorithme ne pourra pas non plus générer un positionnement des points équivalent à celui présenté à la figure 1.3 (**C**).

Les données observées dans la nature exhibant très souvent des caractéristiques hautement non linéaires, il apparaît impératif de développer des techniques de réduction de dimension pouvant apprendre les caractéristiques de variétés non linéaires de plus faible dimension observées dans des espaces de haute dimension en ne considérant qu'un échantillon fini de données. De tels algorithmes sont présentés au chapitre 4.

CHAPITRE 4

Méthodes non linéaires de réduction de dimension

Certains algorithmes d'apprentissage linéaires ne font intervenir que des produits scalaires entre les observations sans jamais considérer ces dernières individuellement. Comme il a été mentionné au chapitre 2, ces algorithmes peuvent être généralisés afin de considérer les relations entre des fonctions non linéaires des observations. Il suffit de remplacer chaque produit scalaire $\langle x, y \rangle$ par un noyau $k(x, y)$ dans l'algorithme. Le choix de k permettra alors de considérer des produits scalaires (ou relations de covariance) dans un espace de caractéristiques possiblement non linéaires plutôt que dans l'espace original d'observation.

Si elle est utilisée judicieusement, cette approche permet de résoudre les défiences des méthodes linéaires exposées à la section 3.3.

4.1 ACP à noyau

L'idée de réaliser une application implicite dans un espace de caractéristiques de plus haute dimension fut très féconde dans le contexte des machines à vecteur de support (VAPNIK 1998). Il a donc été naturel de se demander si elle pouvait être appliquée à d'autres algorithmes d'apprentissage.

On remarque que l'ACP présentée à la section 3.1 ne fait intervenir que des produits scalaires dans le calcul de la matrice de covariance des exemples et ne considère jamais chaque échantillon en particulier. (SCHÖLKOPF, SMOLA et MÜLLER 1998) proposèrent donc d'appliquer les résultats obtenus au chapitre 2 à la méthode d'analyse en composantes principales exposée à la section 3.1.

Il sera par conséquent possible d'obtenir les composantes principales de fonctions non linéaires de l'espace d'observation. Ces fonctions pourront par exemple être des corrélations d'ordre supérieur entre les variables observées. La présentation de l'algorithme est similaire à celle proposée par (SCHÖLKOPF, SMOLA et MÜLLER 1998).

Soit un ensemble d'observations centrées $\boldsymbol{x}_k \in \mathbb{R}^n, k = 1, \ldots, m$ avec $\sum_{k=1}^{m} \boldsymbol{x}_k = 0$. L'équation (3.2) donne la matrice de covariance des observations

$$C = \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{x}_j \boldsymbol{x}_j^T.$$

Rappelons que l'ACP résoud l'équation

$$\lambda \mathbf{v} = C\mathbf{v} \tag{4.1}$$

pour des valeurs propres $\lambda \geq 0$ (C est semi définie positive (LANG 1989)) et des vecteurs propres $\mathbf{v} \in \mathbb{R}^n \setminus \{0\}$. Comme

$$\lambda \mathbf{v} = C\mathbf{v} = \frac{1}{m} \sum_{j=1}^{m} \langle \boldsymbol{x}_j, \mathbf{v} \rangle \boldsymbol{x}_j,$$

toutes les solutions $\mathbf{v}$ avec $\lambda \neq 0$ doivent se trouver dans le sous-espace généré

par $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$. Ainsi, (4.1) est équivalent à

$$\lambda \langle \boldsymbol{x}_k, \mathbf{v} \rangle = \langle \boldsymbol{x}_k, C\mathbf{v} \rangle \ \forall k = 1, \ldots, m.$$

L'ACP à noyau réalise le même algorithme dans un espace de caractéristiques $\mathcal{H}$ associé à un produit scalaire. Cet espace est relié à l'espace d'observation par une application possiblement non linéaire

$$\tilde{\Phi} : \mathbb{R}^n \to \mathcal{H}, \boldsymbol{x} \mapsto \mathbf{X}$$

où $\mathcal{H}$ est de dimension arbitrairement grande, possiblement infinie et $\mathbf{X} = \tilde{\Phi}(\boldsymbol{x})$. Cette application est implicitement définie par un noyau k, tel qu'explicité au chapitre 2.

Le théorème 3.1 présuppose que les observations $\boldsymbol{x}_k$ sont centrées, c'est-à-dire que $\sum_i \boldsymbol{x}_k = 0$. Cette contrainte est aussi nécessaire dans l'espace de caractéristiques $\mathcal{H}$ pour pouvoir y effectuer l'ACP. On définit par conséquent

$$\Phi(\boldsymbol{x}_i) := \tilde{\Phi}(\boldsymbol{x}_i) - \frac{1}{n} \sum_{i=1}^{m} \tilde{\Phi}(\boldsymbol{x}_i),$$

ce qui donne bien $\sum_i \Phi(\boldsymbol{x}_i) = 0$.

Soit $\tilde{k}$ le noyau déterminant implicitement $\tilde{\Phi}$. Pour déterminer $\Phi(x)$, il suffit de définir le noyau associé k tel que

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}) &:= \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle \\ &= \langle (\tilde{\Phi}(\boldsymbol{x}) - E_{\boldsymbol{x}'}[\tilde{\Phi}(\boldsymbol{x}')]), (\tilde{\Phi}(\boldsymbol{y}) - E_{\boldsymbol{y}'}[\tilde{\Phi}(\boldsymbol{y}')]) \rangle \\ &= \tilde{k}(x, y) - E_{\boldsymbol{x}'}[\tilde{k}(\boldsymbol{x}', \boldsymbol{y})] - E_{\boldsymbol{y}'}[\tilde{k}(\boldsymbol{x}, \boldsymbol{y}')] + E_{\boldsymbol{x}'}[E_{\boldsymbol{y}'}[\tilde{k}(\boldsymbol{x}', \boldsymbol{y}')]]. \end{aligned}$$

Dans $\mathcal{H}$, la matrice de covariance empirique est donnée par

$$\bar{C} = \frac{1}{m} \sum_{j=1}^{m} \Phi(\boldsymbol{x}_j) \Phi(\boldsymbol{x}_j)^T.$$

Si $\mathcal{H}$ est de dimension infinie, on peut concevoir $\Phi(\boldsymbol{x}_j)\Phi(\boldsymbol{x}_j)^T$ comme un

opérateur linéaire définissant l'application

$$\mathbf{X} \mapsto \Phi(\boldsymbol{x}_j)\langle \Phi(\boldsymbol{x}_j), \boldsymbol{x} \rangle,$$

le produit scalaire étant bien défini dans un espace de Hilbert. On peut maintenant trouver les valeurs propres $\lambda \geq 0$ et les vecteurs propres $\mathbf{V} \in \mathcal{H} \setminus \{0\}$ satisfaisant

$$\lambda \mathbf{V} = \bar{C}\mathbf{V}.$$

Encore une fois, toutes les solutions $\mathbf{V}$ avec $\lambda \neq 0$ se trouvent dans le sous-espace généré par $\Phi(\boldsymbol{x}_1), \ldots \Phi(\boldsymbol{x}_m)$. Deux conséquences utiles en découlent. Tout d'abord, on peut considérer les équations

$$\lambda \langle \Phi(\boldsymbol{x}_k), \mathbf{V} \rangle = \langle \Phi(\boldsymbol{x}_k), \bar{C}\mathbf{V} \rangle \ \forall k = 1, \ldots, m. \tag{4.2}$$

Deuxièmement, il existe des coefficients $\alpha_i (i = 1, \ldots, m)$ tels que

$$\mathbf{V} = \sum_{i=1}^{m} \alpha_i \Phi(\boldsymbol{x}_i). \tag{4.3}$$

En combinant (4.2) et (4.3), on obtient

$$\lambda \sum_{i=1}^{m} \alpha_i \langle \Phi(\boldsymbol{x}_k), \Phi(\boldsymbol{x}_i) \rangle = \frac{1}{m} \sum_{i=1}^{m} \alpha_i \left\langle \Phi(\boldsymbol{x}_k), \sum_{j=1}^{m} \Phi(\boldsymbol{x}_j) \langle \Phi(\boldsymbol{x}_j), \Phi(\boldsymbol{x}_i) \rangle \right\rangle \tag{4.4}$$

pour tout $k = 1, \ldots, m$. Si on définit la matrice carrée K de dimension m par

$$K_{ij} := \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle,$$

alors (4.4) se lit

$$m\lambda K \boldsymbol{\alpha} = K^2 \boldsymbol{\alpha}, \tag{4.5}$$

où $\boldsymbol{\alpha}$ est le vecteur colonne avec entrées $\alpha_1, \ldots, \alpha_m$. (SCHÖLKOPF, SMOLA et MÜLLER 1998) montrent que les solutions de (4.5) sont les mêmes que les solutions de

$$m\lambda \boldsymbol{\alpha} = K \boldsymbol{\alpha} \tag{4.6}$$

pour les valeurs propres non nulles.

Soient $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ les valeurs propres de K solutions de (4.6) et $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m$ les vecteurs propres correspondants. Soit λ_p la dernière valeur propre non nulle. On normalise $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m$ afin que les vecteur propres correspondants dans $\mathcal{H}$ soient normalisés, c'est-à-dire que

$$\langle \mathbf{V}^k, \mathbf{V}^k \rangle = 1 \text{ pour tout } k = 1, \ldots, p.$$

Par (4.3) et (4.5), on constate que cette condition devient

$$\begin{aligned} 1 &= \textstyle\sum_{i,j=1}^{m} \alpha_i^k \alpha_j^k \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle &&= \textstyle\sum_{i,j=1}^{m} \alpha_i^k \alpha_j^k K_{ij} \\ &= \langle \boldsymbol{\alpha}^k, K\boldsymbol{\alpha}^k \rangle &&= \lambda_k \langle \boldsymbol{\alpha}^k, \boldsymbol{\alpha}^k \rangle. \end{aligned}$$

L'analyse en composantes principales permet de calculer la projection d'un nouveau point $\boldsymbol{x}$ d'image $\Phi(x)$ dans l'espace caractéristique $\mathcal{H}$ sur les vecteurs propres $\mathbf{V}^k (k = 1, \ldots, p)$ de cet espace par

$$\langle \mathbf{V}^k, \Phi(\boldsymbol{x}) \rangle = \sum_{i=1}^{m} \alpha_i^k \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}) \rangle. \tag{4.7}$$

La projection d'un nouveau point dans l'espace de caractéristiques peut donc se faire avec un coût linéaire en fonction de m. L'équation (4.7) joue un rôle fondamental dans la généralisation des algorithmes de réduction de dimension présentée au chapitre 5. (Williams et Seeger 2000) ont noté que cette projection est proportionnelle à la formule de Nyström (Baker 1977), qui a été utilisée pour prédire les valeurs d'un vecteur propre pour une nouvelle observation de façon à accélérer les méthodes à noyaux en trouvant les vecteurs propres pour un sous-ensemble des données (Williams 2001).

L'algorithme d'analyse en composantes principales à noyau est détaillé au tableau 4.1. En utilisant un noyau correspondant aux conditions du théorème 2.2, on constate que cette procédure revient à effectuer une analyse en composantes principales dans un espace de caractéristiques de haute dimension. Toutes les propriétés des projections linéaires produites par l'ACP classique seront partagées dans l'espace caractéristique par celles produites par l'ACP

1. Calculer $K_{ij} = (k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{ij}$;
2. Trouver la solution de (4.5) donnant les d valeurs propres et vecteurs propres principaux de K et normaliser ces vecteurs propres avec les contraintes $\lambda_k \langle \boldsymbol{\alpha}^k, \boldsymbol{\alpha}^k \rangle = 1$.

Tableau 4.1 – *Analyse en composantes principales à noyau permettant d'exprimer les données en d dimensions.*

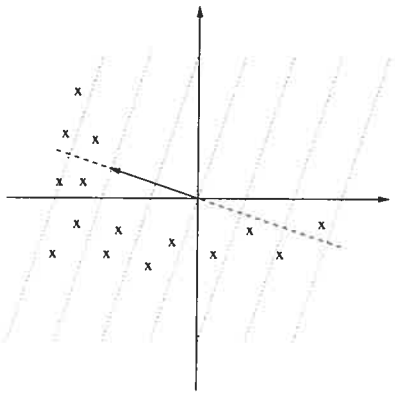

Figure 4.1 – *Axe linéaire principal des données. Les lignes pointillées perpendiculaires à l'axe principal correspondent aux lignes de projection constante sur l'axe principal.*

à noyau.

La figure 4.1 montre l'axe principal d'une configuration fictive de points qui serait générée par l'analyse en composantes principales classique. Cette projection est équivalente à la projection donnée par l'analyse en composantes principales à noyau si on choisit $k(x, y) = \langle x, y \rangle$. On voit à la figure 4.2 que pour l'analyse en composantes principales à noyau, les lignes de projection constante dans l'espace de caractéristiques sont non linéaires dans l'espace d'observation, pour les mêmes données qu'à la figure 4.1.

L'analyse en composantes principales à noyau fournit un plongement pour des points ne faisant pas partie des observations d'entraînement grâce à l'équation (4.7). Les autres algorithmes présentés dans ce chapitre n'offrent pas une telle généralisation aux points hors échantillon. La contribution essentielle de

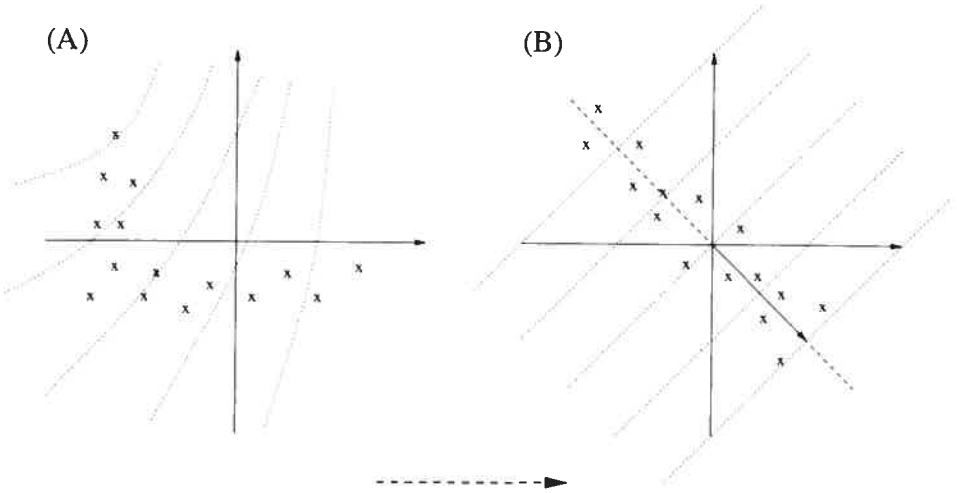

Figure 4.2 – *(A) Pour les même données qu'à la figure 4.1, les lignes pointillées correspondent aux lignes de projection constante dans l'espace de caractéristiques. (B) Dans l'espace de caractéristiques, les points sont projetés linéairement sur l'axe principal.*

ce mémoire consiste donc à présenter, au chapitre 5, un moyen d'estimer les coordonnées de points hors échantillon projetées sur les composantes principales de l'espace de caractéristiques induit par le noyau présent dans chacun de ces algorithmes.

4.2 Isomap

Isomap (TENENBAUM, SILVA et LANGFORD 2000) possède l'efficacité computationnelle, l'optimalité globale et la garantie de convergence asymptotique de l'ACP et de MDS tout en étant capable d'apprendre une grande classe de variétés non linéaires.

Isomap est une généralisation de MDS (présenté à la section 3.2) qui préserve la géométrie intrinsèque des données, capturée par les longueurs des géodésiques passant par la variété à partir de laquelle sont tirées les données. Pour des points voisins, les distances euclidiennes dans l'espace d'observation sont de bonnes approximations des longueurs des géodésiques. On construit un graphe reliant chaque point à ses k plus proches voisins. Les longueurs des géodésiques entre deux points éloignés sont alors estimées en trouvant la longueur du plus court chemin entre ces deux points dans le graphe. Il suffit finalement d'appliquer MDS aux distances obtenues pour obtenir un positionnement des points dans un espace de dimension réduite.

Les trois étapes fondamentales d'Isomap sont énumérées au tableau 4.2. La première étape détermine quels points sont voisins sur la variété V en se basant sur les distances $d_{\mathcal{X}}(i,j)$ entre les paires de points i,j dans l'espace d'observation $\mathcal{X}$ afin de construire un graphe G approximant la variété. La seconde étape consiste essentiellement à approximer les longueurs des géodésiques $d_V(i,j)$ entre toutes les paires de points i,j sur la variété. Le calcul du plus court chemin $d_G(i,j)$ fournit cette estimation. L'algorithme proposé à l'étape 2 du tableau 4.2 pour ce faire peut être remplacé par tout autre algorithme de plus court chemin dans un graphe. L'étape finale applique MDS à la matrice de distances D_G afin de fournir un positionnement des observations en d dimensions

1. Définir le graphe G sur l'ensemble des observations en connectant i et j si $d_{\mathcal{X}}(i,j) < \epsilon$ (ϵ-Isomap) ou si i est un des k plus proches voisins de j en fonction de $d_{\mathcal{X}}(i,j)$ (k-Isomap) ;
2. Initialiser $d_G(i,j) = d_{\mathcal{X}}(i,j)$ si i et j sont connectés par un arc et $d_G(i,j) = \infty$ sinon. Pour chaque valeur de $k = 1, 2, \ldots, p$, remplacer toutes les entrées $d_G(i,j)$ par $\min(d_G(i,j), d_G(i,k) + d_G(k,j))$. La matrice $D_G = (d_G(i,j))$ finalement obtenue contient les plus courts chemins entre toutes les paires de points dans G ;
3. Soit λ_s la s-ième valeur propre (en ordre décroissant) de la matrice M définie en (3.4) à partir de D_G et v_s^i la i-ième composante du s-ième vecteur propre. On a finalement que la s-ième composante de $\boldsymbol{y}_i$ est donnée par $\sqrt{\lambda_s} v_s^i$.

Tableau 4.2 – *L'algorithme Isomap prend en entrée les distances $d_{\mathcal{X}}(i,j)$ entre toutes les paires i,j de p points de donnée dans l'espace d'observation de haute dimension $\mathcal{X}$. Cette distance peut être la distance euclidienne usuelle ou toute métrique spécifique à un domaine d'application particulier. L'algorithme produit des vecteurs de sortie $\boldsymbol{y}_i \in \mathbb{R}^m$ qui représentent le mieux possible la géométrie intrinsèque des données. Le seul paramètre libre (ϵ ou k) apparaît à l'étape 1.*

respectant le plus possible les longueurs des géodésiques sur V.

Soit D_Y la matrice des distances euclidiennes définie par $d_Y(i,j) = ||\boldsymbol{y}_i - \boldsymbol{y}_j||$. M_G et M_Y sont construites à partir de D_G et D_Y de la même façon qu'en (3.4). On observe que les vecteurs $\boldsymbol{y}_i$ sont les vecteurs de dimension d qui minimisent la fonction de coût (3.9) ici donnée par

$$||M_G - M_Y||_{L^2}$$

où $||A||_{L^2}$ est la norme matricielle $\sqrt{\sum_{i,j} A_{ij}^2}$.

Pour toutes les paires de points i,j, la distance $d_G(i,j)$ converge asymptotiquement vers $d_V(i,j)$ avec l'augmentation du nombre de points pour une classe de variétés V très étendue, incluant des variétés non euclidiennes (TENENBAUM, SILVA et LANGFORD 2000). La vitesse de convergence dépend entre autres de la courbure de la variété et de la densité des points. Si la densité des points n'est pas uniforme ou si la courbure de la variété est extrêmement

forte, la convergence asymptotique est toujours garantie, mais la taille de l'échantillon nécessaire à l'estimation acceptable des longueurs des géodésiques peut être arbitrairement grande, donc impossible à obtenir en pratique.

4.3 Plongement localement linéaire

Le plongement localement linéaire, ou *locally linear embedding* (LLE) (ROWEIS et SAUL 2000), tente de résoudre le même problème que Isomap par une approche alternative. Chaque point est ici caractérisé par sa reconstruction par ses plus proches voisins.

Encore une fois, les données observées sont constituées de m points de dimension n approximativement situés sur une variété V de dimension $d < n$. Si le nombre de points est suffisamment grand, on peut supposer que chaque point et ses plus proches voisins sont approximativement situés sur une partie localement linéaire de V. En fait, cette géométrie locale peut être caractérisée par des coefficients linéaires de reconstruction de chaque point à partir de ses voisins.

On commence d'abord par identifier les voisins de chaque point. Comme pour Isomap, on peut choisir les k plus proches voisins ou bien sélectionner tous les points dans un voisinage de taille ϵ du point $\boldsymbol{x}_i$. Si le nombre de voisins k est fixe, on doit avoir $k > d$, dans la mesure où k voisins peuvent reconstruire un sous-espace de dimension maximale $k-1$. En pratique, la détermination du voisinage de chaque point constitue l'occasion d'incorporer des connaissances disponibles a priori dans l'algorithme.

On peut mesurer l'erreur de reconstruction d'un point par ses voisins à l'aide de

$$\sum_i |\boldsymbol{x}_i - \sum_j W_{ij}\boldsymbol{x}_j|^2 \tag{4.8}$$

où chaque W_{ij} est le coefficient de reconstruction de $\boldsymbol{x}_i$ à partir de $\boldsymbol{x}_j$.

Afin d'estimer les W_{ij}, on minimise (4.8) avec deux contraintes. Tout d'abord, chaque $\boldsymbol{x}$ n'est reconstruit qu'à partir de ses plus proches voisins.

De cette façon, $W_{ij} = 0$ si $\boldsymbol{x}_j$ n'est pas un voisin de $\boldsymbol{x}_i$. La seconde contrainte consiste à exiger que $\sum_j W_{ij} = 1$. Ainsi, la reconstruction d'un point à partir de ses voisins est invariante pour toute rotation, tout changement d'échelle ou toute translation de ce point et de ses voisins. L'invariance aux translations est une conséquence de la deuxième contrainte.

(Saul et Roweis 2003) montrent que la minimisation de (4.8) équivaut à trouver pour chaque point $\boldsymbol{x}$ une solution au système d'équations linéaires

$$\sum_k G_{jk} w_k = 1$$

où G_{jk} est une matrice locale de Gram donnée par

$$G_{jk} = \langle \boldsymbol{x} - \boldsymbol{\eta}_j, \boldsymbol{x} - \boldsymbol{\eta}_k \rangle \tag{4.9}$$

et $\boldsymbol{\eta}_i$ est le i-ième voisin de $\boldsymbol{x}$. Dans le cas où G_{jk} est singulière ou presque singulière, la matrice G_{jk} doit être régularisée en y ajoutant un petit multiple de la matrice identité.

Étant donné que les données sont approximativement situées sur la variété V, il est possible d'imaginer qu'il existe une application linéaire (constituée d'une translation, d'une rotation et d'un changement d'échelle) qui associe les coordonnées en haute dimension de chaque voisinage d'un point à des coordonnées internes globales de V. On constate que les poids de reconstruction W_{ij} reflètent les propriétés des données qui sont invariantes sous ces transformations. Leur caractérisation de l'espace d'observation demeure donc valide dans des parties localement linéaires de V. Les poids qui reconstruisent un point $\boldsymbol{x}$ en n dimensions devraient donc reconstruire aussi la projection de ce point dans les coordonnées en d dimensions associées à V.

LLE construit une application en plus faible dimension préservant le voisinage basée sur cette idée. Les projections $\boldsymbol{y}_i$ en d dimensions des points $\boldsymbol{x}_i$ sont donc choisies de façon à minimiser le coût de reconstruction des points

dans l'espace de projection donné par

$$\sum_i |\boldsymbol{y}_i - \sum_j W_{ij}\boldsymbol{y}_j|^2. \tag{4.10}$$

Cette fois, les poids W_{ij} sont fixés afin de pouvoir optimiser le plongement déterminé par les projections $\boldsymbol{y}_i$. L'idée est donc de trouver des coordonnées $\boldsymbol{y}_i$ en faible dimension qui sont reconstruites par les mêmes poids W_{ij} que les observations $\boldsymbol{x}_i$ en haute dimension.

On peut écrire l'équation (4.10) sous forme quadratique comme

$$\sum_{ij} \widetilde{M}_{ij}\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$$

où $\widetilde{M}$ est donnée par

$$\widetilde{M}_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki}W_{kj}. \tag{4.11}$$

Comme une translation globale des $\boldsymbol{y}$ n'affecte pas le coût de reconstruction, on peut exiger que

$$\sum_i \boldsymbol{y}_i = 0. \tag{4.12}$$

Afin d'éliminer les degrés de liberté dûs à l'invariance aux rotations et au choix de l'échelle, on peut aussi fixer

$$\frac{1}{m}\sum_i \boldsymbol{y}_i\boldsymbol{y}_i^T = I, \tag{4.13}$$

où I est la matrice identité.

La minimisation de (4.10) sous les contraintes (4.12) et (4.13) peut être réalisée en trouvant les $d+1$ vecteurs propres de $\widetilde{M}$ de plus petites valeurs propres, tel que le stipule le théorème de Rayleitz-Ritz (HORN et JOHNSON 1990). La i-ième composante du j-ième vecteur propre correspond alors à la j-ième composante de la projection de la i-ième observation dans l'espace de plus faible dimension. On élimine le vecteur propre de plus petite valeur

1. Caculer les voisins de chaque observation $\boldsymbol{x}_i$.
2. Calculer les poids W_{ij} qui reconstruisent le mieux possible chaque $\boldsymbol{x}_i$ à partir de ses voisins en minimisant (4.8).
3. Calculer le plongement donné par les $\boldsymbol{y}_i$ qui sont le mieux reconstruits par les W_{ij}, en minimisant l'équation quadratique (4.10) par la solution d'une équation à valeurs propres. On ne conserve que les vecteurs propres de plus petites valeurs propres non nulles.

Tableau 4.3 – *Algorithme de plongement localement linéaire (LLE).*

propre, car il correspond à une translation et sa valeur propre est nulle. Toutes les composantes de ce vecteur ont la valeur 1. La suppression du vecteur unité permet aux autres $\boldsymbol{y}_i$ d'avoir une moyenne nulle, car ils sont alors orthogonaux à celui-ci. En pratique, il n'est pas nécessaire de calculer explicitement la matrice $\widetilde{M}$ (SAUL et ROWEIS 2003). L'algorithme LLE est résumé au tableau 4.3.

On note que le fait de trouver les vecteurs propres de plus petite valeur propre de $\widetilde{M}$ correspond à trouver les vecteurs propres principaux de $M = cI - \widetilde{M}$. Il est possible de choisir c comme la valeur propre la plus grande de $\widetilde{M}$ de façon à avoir une matrice M semi définie positive. Ainsi, comme le remarque (HAM, LEE et SCHÖLKOPF 2003), LLE peut être vu comme une façon d'accomplir l'ACP à noyau avec une matrice de Gram spécifique. Cette analogie est claire si on note que le fait d'ignorer le vecteur propre principal de M correspond à l'opération de centrage des données effectuée dans l'ACP à noyau.

4.4 Segmentation spectrale

Certaines applications exigent de séparer les observations contenues dans $\mathbb{R}^n$ en groupes de proximité distincts, ou *clusters*, de façon non supervisée et en utilisant une métrique pertinente. Cette tâche est intimement liée à l'apprentissage de variétés dans la mesure où les groupes de données peuvent

1. Poser $\mathbf{c} = (c_1, \ldots, c_K)$ où chaque c_i est un point des données choisi aléatoirement sans remplacement. Les c_i sont appelés les centroïdes de chaque segment ;
2. Associer chaque observaton $\boldsymbol{x}$ au centroïde le plus près ;
3. Changer la position de chaque c_i pour qu'elle corresponde à la moyenne des positions de toutes les observations associées à ce centroïde.
4. Répéter les étapes 2 et 3 jusqu'à ce que tous les changements de position des centroïdes soient inférieurs à un seuil donné.

Tableau 4.4 – *Algorithme K-moyennes.*

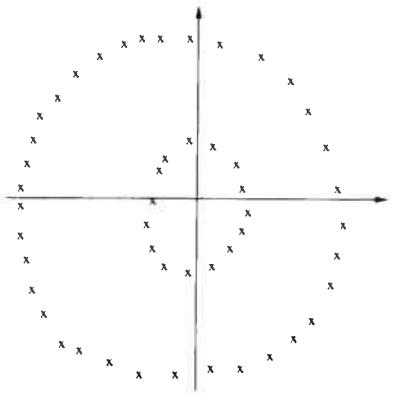

Figure 4.3 – *L'algorithme K-moyennes est incapable de séparer correctement les deux groupes de points.*

être en fait considérés comme des variétés de courbure arbitraire. Les groupes de données et les variétés ne sont en fait que des zones de haute densité.

L'algorithme *K-moyennes*, ou *K-means*, offre une méthode de segmentation simple à implanter présentée au tableau 4.4. Malheureusement, il est clair qu'un tel algorithme se révèle impuissant à séparer les deux groupes de points illustrés à la figure 4.3. En général, l'application de cet algorithme est donc limitée au cas où les données ne sont pas imbriquées ou entrelacées dans l'espace d'observation.

La *segmentation spectrale*, ou *spectral clustering* (Weiss 1999; Ng, Jor-

1. Construire la matrice $\tilde{K}$ telle que $\tilde{K}_{ij} = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\sigma^2)$ si $i \neq j$ et $\tilde{K}_{ii} = 0$. En général, il est possible d'utiliser n'importe quel autre noyau $\tilde{K}$;
2. Construire D la matrice diagonale telle que D_{ii} est la somme des éléments de la i-ième ligne de $\tilde{K}$ et construire la matrice $M = D^{-1/2}\tilde{K}D^{-1/2}$;
3. Trouver $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_d$ les d vecteurs propres de M de plus grandes valeurs propres orthogonaux entre eux et former la matrice $E = [\boldsymbol{e}_1 \boldsymbol{e}_2 \ldots \boldsymbol{e}_d] \in \mathbb{R}^{n \times d}$;
4. Former Y à partir de E en normalisant les vecteurs lignes de E pour qu'ils soient de norme unitaire, c'est-à-dire $Y_{ij} = E_{ij}/(\sum_j E_{ij}^2)^{1/2}$;
5. Les rangées de Y sont considérées comme des points dans $\mathbb{R}^d$. On applique l'algorithme $K-$moyennes à ces points ;
6. On assigne le point $\boldsymbol{x}_i$ au groupe j si et seulement si la ligne i de la matrice Y a été assignée au groupe j.

Tableau 4.5 – *Algorithme de segmentation spectrale.*

DAN et WEISS 2002), permet de s'affranchir des limites inhérentes aux méthodes de segmentation précédemment développées. Basée encore une fois sur le calcul des vecteurs propres d'une matrice de Gram, cette méthode de segmentation a tout d'abord été utilisée en vision artificielle (MALIK, BELONGIE, LEUNG et SHI 2000).

Le tableau 4.5 explicite en détails une façon d'effectuer les différentes étapes de la segmentation spectrale. Cet algorithme permet de réaliser une application de variétés hautement non linéaires vers des sous-espaces linéaires distincts pour chaque variété, tel qu'illustré à la figure 4.4. On constate sur cette figure que les deux groupes de points projetés forment des lignes qui forment approximativement un angle droit par rapport à l'origine. Cette constatation suggère de normaliser les coordonnées de chaque point sur le cercle unité avant de séparer les projections en groupes distincts en utilisant K-moyennes.

La qualité de la segmentation dépend fortement du choix du noyau ou des paramètres le caractérisant, comme la variance σ du noyau gaussien. Ainsi, le

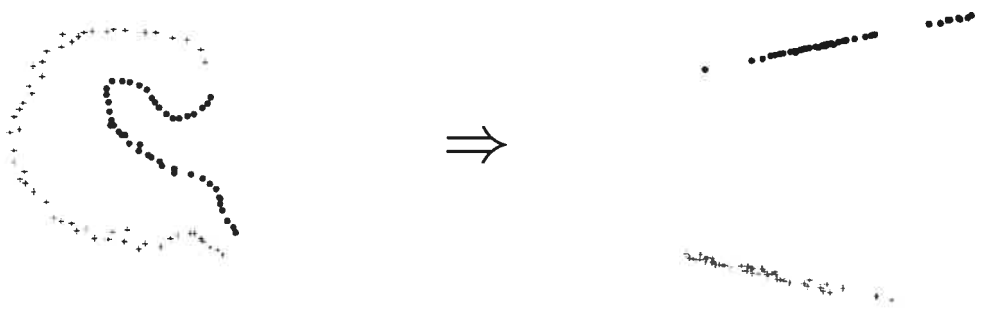

Figure 4.4 – Exemple d'une transformation apprise au cours de la segmentation spectrale. Les observations $\boldsymbol{x}$ sont à gauche et les points obtenus en considérant les lignes de Y sont à droite. Les croix et les cercles ne sont pas utilisés par l'algorithme. L'algorithme sépare les deux groupes et révèle la structure interne des deux variétés ayant généré les points.

noyau gaussien utilisé à l'étape 1 de l'algorithme présenté au tableau 4.5 peut être substitué à tout autre noyau symétrique. Toutefois, le théorème 2.2 exige que le noyau choisi soit défini positif pour garantir la décomposition spectrale dans l'espace de caractéristiques.

Des approches algorithmiques destinées à selectionner des noyaux pour la segmentation spectrale dépendants des données et adaptés à des applications particulières pourraient être développées.

CHAPITRE 5

Généralisation des algorithmes

La plupart des algorithmes d'apprentissage fournissent une fonction qui minimise la moyenne empirique d'un critère de perte auquel est greffé une composante de régularisation (VAPNIK 1995). En d'autres termes, on choisit une fonction dont le coût est minimal par rapport aux données observées. La composante de régularisation sert à imposer des contraintes sur la forme de la fonction afin d'éviter d'obtenir une fonction trop sensible à la variance des données. La fonction ainsi apprise peut alors être appliquée à des points non utilisés pour l'entraînement. La solution idéale est une fonction qui minimise l'espérance du coût par rapport à la distribution inconnue à partir de laquelle les données ont été tirées. Il s'agit de l'erreur de généralisation. Cependant, une telle caractérisation n'était pas disponible pour des algorithmes de plongement spectral comme le positionnement multidimensionel, la segmentation spectrale, le plongement localement linéaire et Isomap. Ces algorithmes ne fournissent pas de fonction pouvant être appliquée à de nouveaux points et la notion d'erreur de généralisation n'est pas clairement définie.

Afin de pallier à ces déficiences, on montre une relation directe entre les méthodes de plongement spectral développées au chapitre précédent et l'analyse en composantes principales à noyau. Les deux approches sont en fait des

cas particuliers d'un problème d'apprentissage plus général, qui consiste à apprendre les fonctions propres principales d'un opérateur linéaire défini à partir d'un noyau et d'une densité. Les méthodes spectrales présentées ne fournissant qu'un plongement pour les points d'entraînement, cette analyse donne lieu à une généralisation simple des algorithmes de segmentation spectrale, MDS, LLE et Isomap aux exemples hors échantillon. L'analyse proposée fournit pour chacun de ces algorithmes la définition d'une fonction de coût dont la moyenne empirique est minimisée par les algorithmes originaux et l'espérance de ce coût définit une performance de généralisation qui indique clairement ce qui est appris par ces algorithmes asymptotiquement. Des expériences effectuées avec LLE, Isomap, MDS et la segmentation spectrale montrent que les perturbations du plongement des exemples de l'ensemble d'entraînement dûs au remplacement de quelques exemples par d'autres tirés de la même distribution sont de grandeur comparables aux différences entre le plongement estimé par l'extension hors échantillon proposée et le plongement qui aurait été obtenu en incluant ces exemples dans l'ensemble d'entraînement.

5.1 Noyaux dépendants des données

Tous les algorithmes de segmentation spectrale présentés construisent une matrice de similitude M de taille $m \times m$, calculent leurs vecteurs propres principaux v_k et leurs valeurs propres associées l_k et associent au i-ième exemple d'entraînement un plongement de coordonnées $(v_{1i}, v_{2i}, \ldots)$ pour la segmentation spectrale et LLE et un plongement de coordonnées $(\sqrt{l_1}v_1, \sqrt{l_2}v_2, \ldots)$ pour Isomap et MDS. En général, M_{ij} ne dépend pas seulement de $\boldsymbol{x}_i$ et $\boldsymbol{x}_j$, mais plutôt de tous les exemples de l'ensemble d'entraînement. Il est cependant possible d'écrire $M_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ où k est vu comme un noyau dépendant des données.

Il importe de définir pour chacun de ces algorithmes un noyau k pouvant être appliqué à de nouveaux points hors échantillon de façon à pouvoir appliquer la formule de Nyström à ces points pour obtenir leur projection sur les

axes principaux des données dans l'espace de caractéristiques, tel qu'explicité à la section 5.2.

5.1.1 Noyau généralisé pour MDS

Dans le cas de l'algorithme MDS, on peut concevoir la matrice M du théorème 3.2 comme une matrice de Gram construite à partir d'un noyau k dépendant des données défini comme

$$k(\boldsymbol{x}_r, \boldsymbol{x}_s) = -\frac{1}{2}\left(d_{rs}^2 - \frac{1}{n}\sum_{j=1}^{m} d_{\cdot j}^2 - \frac{1}{n}\sum_{i=1}^{m} d_{i\cdot}^2 + \frac{1}{n^2}\sum_{i,j=1}^{m} d_{ij}^2\right) \tag{5.1}$$

où $d_{\cdot\cdot}^2$ est défini à l'équation (3.6). En fait, k ne dépend pas seulement de $\boldsymbol{x}_r$ et $\boldsymbol{x}_s$, mais aussi de toutes les autres observations de l'ensemble d'entraînement. On constate que le noyau k peut très bien être appliqué aux points hors échantillon.

Une généralisation de MDS aux points hors échantillon a été proposée dans (Gower 1968), où on trouve une solution exacte pour les coordonnées d'un nouveau point. Ces coordonnées doivent être cohérentes avec les distances aux points de l'ensemble d'entraînement, ce qui nécessite en général l'ajout d'une dimension supplémentaire.

L'approche proposée à la section 5.2 consiste plutôt à appliquer k à des points hors échantillon, de façon à pouvoir appliquer la formule de Nyström pour obtenir leur projection sur les axes principaux dans l'espace de caractéristique.

5.1.2 Noyau généralisé pour la segmentation spectrale

De la même façon qu'en (5.1) pour l'algorithme MDS, on peut considérer la matrice M de l'étape 2 du tableau 4.5 comme une matrice de Gram formée à partir du noyau

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{n\tilde{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{\sum_{k=1}^{m} \tilde{K}(\boldsymbol{x}_k, \boldsymbol{x}_i) \sum_{l=1}^{m} \tilde{K}(\boldsymbol{x}_j, \boldsymbol{x}_l)}}. \tag{5.2}$$

Il s'agit d'une simple réécriture des deux premières étapes de l'algorithme. Encore une fois, ce noyau k dépend de tous les exemples de l'ensemble d'entraînement et peut être appliqué à des exemples hors échantillon.

5.1.3 Noyau généralisé pour Isomap

Il existe plusieurs façons de définir un noyau généralisant à de nouveaux points le noyau k définissant la matrice de Gram M intervenant dans l'algorithme Isomap. L'approche adoptée ici consiste à ne considérer un point hors échantillon dans le calcul des longueurs des géodésiques que si ce point est situé à l'*extrémité* d'une géodésique. Cette approche est illustrée à la figure 5.1.

Soit $d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{y})$ la distances entre les points $\boldsymbol{x}$ et $\boldsymbol{y}$ de l'espace d'observation et $d_G(\boldsymbol{x}, \boldsymbol{y})$ la longueur de la géodésique entre les points $\boldsymbol{x}$ et $\boldsymbol{y}$ préalablement calculée par Isomap comme au tableau 4.2. Soit $D = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ un ensemble d'observations fournies en entrée à l'algorithme Isomap et $\mathcal{N}(\boldsymbol{x})$ l'ensemble des k plus proches voisins dans D de $\boldsymbol{x}$. On définit le noyau généralisé $k(\boldsymbol{x}, \boldsymbol{y})$ par

$$k(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} d_G(\boldsymbol{x}, \boldsymbol{y}) & \text{si } \boldsymbol{x} \in D \text{ et } \boldsymbol{y} \in D \\ \min_{\mathbf{z} \in \mathcal{N}(\boldsymbol{x})}(d_{\mathcal{X}}(\boldsymbol{x}, \mathbf{z}) + d_G(\mathbf{z}, \boldsymbol{y})) & \text{si } \boldsymbol{x} \notin D \text{ et } \boldsymbol{y} \in D \\ \min_{\mathbf{z} \in \mathcal{N}(\boldsymbol{y})}(d_G(\boldsymbol{x}, \mathbf{z}) + d_{\mathcal{X}}(\mathbf{z}, \boldsymbol{y})) & \text{si } \boldsymbol{x} \in D \text{ et } \boldsymbol{y} \notin D. \end{cases}$$

Il n'est pas nécessaire de définir $k(\boldsymbol{x}, \boldsymbol{y})$ pour le cas où $\boldsymbol{x} \notin D$ et $\boldsymbol{y} \notin D$ dans ce contexte.

Les valeurs de k pour les points de l'ensemble d'entraînement sont conformes aux valeurs de M obtenues a priori. Cette méthode respecte le principe généralement appliqué dans les algorithmes d'apprentissage consistant à ne pas modifier les paramètres d'un algorithme lors de son utilisation sur des points hors échantillon. Par exemple, les poids d'un réseau de neurones ne sont pas modifiés par son utilisation sur un ensemble de test.

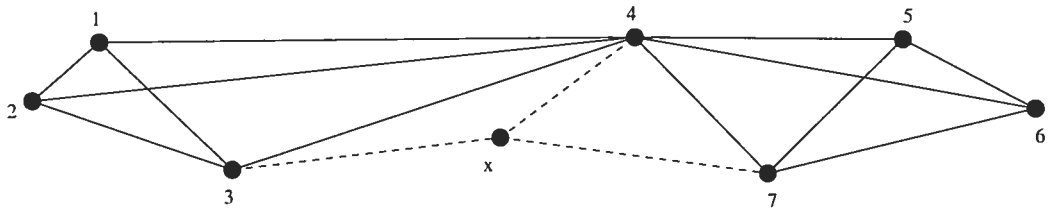

Figure 5.1 – *Exemple d'un graphe généré par Isomap. On suppose que les points 1 à 7 font partie de l'ensemble d'entraînement et sont connectés aux trois plus proches voisins. On connecte le point hors échantillon x aux trois plus proches voisins pour pouvoir calculer $k(x, \cdot)$. Cependant, l'ajout de x dans le graphe ne modifie pas le calcul de $k(3, 7)$, même si la distance entre les points 3 et 7 diminuerait en passant par le point x.*

5.1.4 Noyau généralisé pour LLE

On peut concevoir un noyau généralisé associé à l'algorithme LLE en définissant une matrice de Gram locale de taille $K \times K$ par

$$C(\boldsymbol{x})_{ij} = \langle \boldsymbol{x} - \boldsymbol{x}_{n(x,i)}, \boldsymbol{x} - \boldsymbol{x}_{n(x,j)} \rangle$$

où K est le nombre de plus proches voisins considérés par LLE et $n(x, i)$ est l'indice au sein des données du i-ième plus proche voisin de $\boldsymbol{x}$. Cette matrice correspond à la matrice G de l'algorithme LLE donnée en (4.9). Dans le cas où $C(\boldsymbol{x})$ est singulière ou presque singulière, $C(\boldsymbol{x})$ doit être régularisée en y ajoutant un petit multiple de la matrice identité. On peut alors définir une fonction $w(\boldsymbol{x}_i, \boldsymbol{x}_j)$ asymétrique générant l'équivalent de la matrice W définie en (4.8) par

$$w(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \frac{\sum_q C^{-1}(\boldsymbol{x})_{jq}}{\sum_{pq} C^{-1}(\boldsymbol{x})_{pq}} & \text{avec } j \text{ tq } \boldsymbol{x}_{n(x,j)} = \boldsymbol{y} \text{ si } \boldsymbol{y} \in \mathcal{N}(\boldsymbol{x}) \\ 0 & \text{sinon} \end{cases} . \tag{5.3}$$

où $\mathcal{N}(\boldsymbol{x})$ est l'ensemble des K plus proches voisins de $\boldsymbol{x}$ dans les données observées. Les valeurs de $w(\boldsymbol{x}, \boldsymbol{y})$ pour différents points $\boldsymbol{y}$ de l'ensemble d'entraînement correspondent aux coefficients de reconstruction du point $\boldsymbol{x}$ par les points $\boldsymbol{y}$. Ainsi, ces coefficients correspondent aux W_{ij} trouvés en (4.8)

dans l'algorithme LLE. On constate que $w(\boldsymbol{x}, \boldsymbol{y})$ est non nul si $\boldsymbol{y}$ est l'un des K plus proches voisins de $\boldsymbol{x}$. On a bien que $\sum_i w(\boldsymbol{x}, \boldsymbol{x}_i) = 1$ et que $((\sum_i w(\boldsymbol{x}, \boldsymbol{x}_i)\boldsymbol{x}_i) - \boldsymbol{x})^2$ est minimisé, tel que requis. Seuls les points d'entraînement sont utilisés pour reconstruire un point hors échantillon $\boldsymbol{x}$, mais pas l'inverse. On a bien que $W_{ij} = w(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Le noyau généralisé peut ainsi être exprimé par

$$k(\boldsymbol{x}, \boldsymbol{y}) = w(\boldsymbol{x}, \boldsymbol{y}) + w(\boldsymbol{y}, \boldsymbol{x}) - \sum_{i=1}^{m} w(\boldsymbol{x}_i, \boldsymbol{x}) w(\boldsymbol{x}_i, \boldsymbol{y})$$

comme en (4.11). Lorsque $\boldsymbol{x}$ et $\boldsymbol{y}$ sont absents de l'ensemble d'entraînement, on a que $k(\boldsymbol{x}, \boldsymbol{y}) = 0$. D'autre part, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = M_{ij}$ dans le cas où $\boldsymbol{x}_i$ et $\boldsymbol{x}_j$ sont tous les deux éléments de l'ensemble d'entraînement. Finalement, si seul $\boldsymbol{x}_i$ fait partie des données d'entraînement, $k(\boldsymbol{x}, \boldsymbol{x}_i) = w(\boldsymbol{x}, \boldsymbol{x}_i) = w(\boldsymbol{x}_i, \boldsymbol{x}) = k(\boldsymbol{x}_i, \boldsymbol{x})$, donc k est symétrique.

On dispose donc, pour chaque algorithme spectral présenté, d'un noyau k dépendant des données pouvant générer la matrice de Gram M associée à cet algorithme et pouvant aussi être appliqué à des points hors échantillon. Ces noyaux pourront alors être utilisés à la section suivante afin de pouvoir approximer les projections de points hors échantillon sur les composantes principales de l'espace de caractéristiques.

5.2 Fonctions propres d'un noyau de similarité

Soit un ensemble de données $D = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. Le plongement obtenu à l'aide d'une méthode à noyau converge lorsque le nombre de points de D tend vers l'infini pour les valeurs propres non répétées de la matrice de Gram (Baker 1977; Williams et Seeger 2000; Shawe-Taylor et Williams 2003). Une conjecture peut alors être tentée, soit que chaque vecteur propre converge en fait vers les fonctions propres d'un opérateur linéaire. En d'autres

termes, cela signifie que le i-ième élément du k-ième vecteur propre converge vers l'application de la k-ième fonction propre de cet opérateur à l'observation $\boldsymbol{x}_i$.

Soit un noyau k_n borné de spectre discret, pouvant donc être exprimé (théorème 2.2) par

$$k_n(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{\infty} \alpha_k \phi_k(\boldsymbol{x}) \phi_k(\boldsymbol{y}).$$

On considère l'espace $\mathcal{H}_p$ des fonctions continues définies partout et dont le carré est intégrable avec

$$\int f^2(\boldsymbol{x}) p(\boldsymbol{x}) dx < \infty$$

où $p(x)$ est la densité générant les données. Il existe dans cet espace des classes d'équivalence entre fonctions. Deux fonctions f et g sont équivalentes si et seulement si $\int (f(x) - g(x))^2 p(x) dx = 0$. Toutefois, si p est strictement positif, chaque classe d'équivalence ne contient qu'une fonction presque partout.

On peut alors définir l'opérateur linéaire dont les fonctions propres généralisent les vecteurs propres d'une matrice de Gram. On considère que k_n converge uniformément en probabilité vers une limite k si $n \to \infty$. On associe alors à chaque k_n un opérateur linéaire G_n et à k un opérateur G définis par

$$G_n f = \frac{1}{n} \sum_{i=1}^{n} k_n(\cdot, \boldsymbol{x}_i) f(\boldsymbol{x}_i)$$

et

$$Gf = \int k(\cdot, \boldsymbol{y}) f(\boldsymbol{y}) p(\boldsymbol{y}) d\boldsymbol{y}. \tag{5.4}$$

Ces deux notions sont bien définies car les fonctions f sont définies partout.

La formule de Nyström (BAKER 1977) est donnée par

$$f_{k,n}(\boldsymbol{x}) = \frac{\sqrt{n}}{\lambda_k} \sum_{i=1}^{n} v_{ik} k_n(\boldsymbol{x}, \boldsymbol{x}_i) \tag{5.5}$$

où λ_k et v_{ik} sont respectivement la k-ième valeur propre et la i-ième compo-

sante du k-ième vecteur propre de la matrice de Gram associée au noyau k_n. La formule de Nyström est donc proportionnelle à l'équation (4.7) donnant la projection d'un nouveau point dans l'ACP à noyau sur les composantes principales dans l'espace de caractéristiques associé au noyau.

Soit f_k et $f_{k,n}$ les fonctions propres satisfaisant

$$Gf_k = \lambda_k f_k \tag{5.6}$$

et

$$G_n f_{k,n} = \lambda_{k,n} f_{k,n} \tag{5.7}$$

où λ_k et $\lambda_{k,n}$ sont les valeurs propres correspondantes. On note que si (5.7) n'est évaluée que sur les $\boldsymbol{x}_i \in D$, l'ensemble des équations possibles est réduit au système

$$M_n v_k = n\lambda_{k,n} v_k$$

où M_n est la matrice de Gram générée par le noyau associé à l'opérateur G_n.

On peut montrer que la formule de Nyström donne les fonctions propres de G_n, dont la valeur sur les observations d'entraînement correspond au plongement spectral et que les fonctions propres convergent en fait vers les fonctions propres de G selon certaines conditions. Le théorème suivant (Bengio, Delalleau, Leroux, Paiement, Vincent et Ouimet 2003) caractérise les fonctions propres de G_n, même dans le cas où les valeurs propres sont négatives.

Théorème 5.1 *G_n possède $m < n$ fonctions propres de la forme*

$$f_{k,n}(\boldsymbol{x}) = \frac{\sqrt{n}}{l_k} \sum_{i=1}^{n} v_{ik} k_n(\boldsymbol{x}, \boldsymbol{x}_i) \tag{5.8}$$

pour des valeurs propres $\lambda_{k,n} = l_k/n$ correspondantes non nulles, où $v_k = (v_{1k}, \ldots, v_{nk})^T$ est le k-ième vecteur propre de la matrice de Gram M, associé à la valeur propre l_k.

Pour $\boldsymbol{x}_i \in D$ ces fonctions coincïdent avec les vecteurs propres correspondants. Ainsi, $f_{k,n}(\boldsymbol{x}_i) = \sqrt{n} v_{ik}$.

Preuve On montre tout d'abord que les $f_{n,k}$ coincïdent avec les vecteurs propres de M aux points $\boldsymbol{x}_i \in D$. Pour les $f_{k,n}$ associées aux valeurs propres non nulles,

$$\begin{aligned} f_{k,n}(\boldsymbol{x}_i) &= \frac{\sqrt{n}}{\ell_k} \sum_{j=1}^{n} v_{jk} k_n(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &= \frac{\sqrt{n}}{\ell_k} \lambda_{k,n} v_{ik} \\ &= \sqrt{n} v_{ik}. \end{aligned} \tag{5.9}$$

Les v_k étant orthonormés, les $f_{k.n}$ sont alors différentes entres elles pour des valeurs différentes de k.

Ainsi, pour tout $\boldsymbol{x}$,

$$(G_n f_{k,n})(x) = \frac{1}{n} \sum_{i=1}^{n} k(x, x_i) f_{k,n}(x_i) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} k(x, x_i) v_{ik} = \frac{\ell_k}{n} f_{k,n}(x), \tag{5.10}$$

ce qui montre que f_k est une fonction propre de G_n associée à la valeur propre $\lambda_{k,n} = \ell_k / n$. ■

Ainsi, l'application de la formule de Nyström aux observations d'entraînement revient à calculer les vecteurs propres de la matrice de Gram M. D'autre part, ces résultats montrent que la formule de Nyström peut généraliser un plongement spectral à des points absents de l'ensemble d'entraînement, comme c'est le cas pour la projection (4.7) des points hors échantillon sur les composantes principales des données dans l'espace de caractéristiques dans l'ACP à noyau. En fait, si k est positif semi défini, on peut immédiatement voir l'application de la fonction $f_{k,n}$ donnée par (5.8) comme la projection d'un point de test sur la k-ième composante principale des données dans l'espace de caractéristiques telle que calculée lors de l'ACP à noyau, comme le suggéraient déjà (Williams et Seeger 2000). (Ham, Lee et Schölkopf 2003) ont déjà noté que Isomap et LLE peuvent être interprétés comme une forme d'ACP à noyau sans proposer de méthode de généralisation à des points hors échantillon.

Des analyses de la convergence de l'erreur de généralisation de l'ACP à

noyau (SHAWE-TAYLOR, CRISTIANINI et KANDOLA 2002; SHAWE-TAYLOR et WILLIAMS 2003) tendent à démontrer que les méthodes spectrales estiment en fait la limite convergente des vecteurs propres principaux d'une matrice de Gram. On peut donc concevoir les vecteurs propres comme des estimateurs des fonctions propres, qui peuvent alors être appliquées à de nouveaux points. Les algorithmes de plongement spectral deviennent alors des algorithmes d'induction de fonction.

La formule de Nyström est connue depuis plusieurs années (BAKER 1977). Cette fonction a été utilisée précédemment pour estimer les extensions de vecteurs propres dans une régression par processus gaussien (WILLIAMS et SEEGER 2000) et elle correspond aussi à la projection d'un point de test calculée par l'ACP à noyau à l'équation (4.7).

On peut appliquer la formule de Nyström à des points absents de l'ensemble d'entraînement en utilisant les noyaux généralisés (BENGIO, PAIEMENT et VINCENT 2003) définis à la section 5.1, ce qui revient à généraliser les algorithmes Isomap, LLE, segmentation spectrale et MDS. Ainsi, il est possible d'estimer à l'aide de cette formule le plongement obtenu pour un point hors échantillon sans avoir à recommencer l'exécution de l'algorithme sur l'ensemble d'entraînement auquel on aurait ajouté le nouveau point. Les champs d'application de ces algorithmes se trouvent par conséquent considérablement élargis, dans la mesure où il devient possible de les utiliser sur les points hors échantillon avec une efficacité proportionnelle au nombre de points de l'ensemble d'entraînement.

Des mesures empiriques de la précision de la projection d'un nouveau point obtenue à l'aide de la formule de Nyström sont présentées à la section 5.4.

Étant donné que plusieurs généralisations des vecteurs propres d'une matrice de Gram sont possibles, le résultat suivant, démontré dans (BENGIO, DELALLEAU, LEROUX, PAIEMENT, VINCENT et OUIMET 2003), s'applique à décrire la convergence des fonctions propres avec l'augmentation du nombre d'observations.

Théorème 5.2 *Si le noyau dépendant des données k_n converge uniformément en probabilité et si les fonctions propres $f_{k,n}$ de G_n associées à des valeurs*

propres non nulles convergent uniformément en probabilité, alors leurs limites sont les fonctions propres correspondantes de G.

Ainsi, si le nombre d'observations tend vers l'infini et que les hypothèses sur les noyaux k_n sont respectées, la formule de Nyström donne les "vraies" projections des points sur les composantes principales dans l'espace de caractéristiques. Cependant, la contrainte de convergence uniforme sur les k_n n'a pas été vérifiée pour les noyaux présentés à la section 5.1. Deux pistes de recherche sont alors possibles. D'une part, il est peut-être possible d'alléger les contraintes du théorème 5.2 de façon à éliminer à tout le moins la contrainte d'uniformité sur la convergence des k_n, qui peut s'avérer difficile à respecter en pratique. D'autre part, la convergence uniforme des noyaux définis pourrait être vérifiée.

Dans le cas d'une projection par ACP à noyau effectuée sur un point hors échantillon, les garanties de convergence et de stabilité offertes par le théorème précédent indiquent qu'un vecteur propre principal d'une matrice de covariance empirique estime correctement le vecteur propre correspondant de la vraie matrice de covariance.

5.3 Apprentissage des fonctions propres d'un noyau

Des critères de coût peuvent être défini pour les algorithmes de plongement spectral. Ces critères peuvent être des erreurs de reconstruction dépendant de paires d'observations. La minimisation de leur valeur moyenne produit les vecteurs propres donnant lieu aux plongements calculés par les algorithmes originaux. De plus, leur minimisation sur la vraie distribution ayant généré les données peut éventuellement induire les fonctions propres de l'opérateur linéaire G défini en (5.4).

Dans un autre ordre d'idées, bien que la formule de Nyström donne une solution convergeant asymptotiquement vers la projection d'un point sur les

vraies composantes principales de l'espace de caractéristiques, il peut aussi être pertinent de tenter d'approximer les fonctions propres de G à l'aide d'une méthode d'optimisation de critère de coût par descente de gradient, comme des réseaux de neurones (BISHOP 1995). En effet, si le nombre d'observations est très élevé, la solution d'un système de vecteurs propres peut s'avérer trop coûteuse en pratique ($O(n^2)$). Une méthode d'optimisation en ligne pourrait alors s'avérer une solution alternative avantageuse sur le plan numérique. D'autre part, de par sa nature, la formule de Nyström donne une solution exacte passant par chacune des observations. Une telle approche pourrait donner lieu à du sur-apprentissage par rapport à l'estimation des fonctions propres à l'aide de classes de fonctions plus lisses, comme des réseaux de neurones ou des machines à vecteur de support.

On définit par conséquent des critères dont les extremums sont les fonctions propres à estimer. Les deux théorèmes suivants indiquent que la meilleure approximation de $k(x, y)$ par rapport à la norme de $\mathcal{H}$ en n'utilisant que m termes de la forme $\lambda_k f_k(x) f_k(y)$ est donnée par

$$\sum_{k=1}^{m} \lambda_k f_k(x) f_k(y) \approx k(x, y),$$

où les λ_k sont les valeurs propres en ordre décroissant et les f_k leurs vecteurs propres associés tels que définis en (5.6).

On considère tout d'abord le cas de la fonction propre principale.

Théorème 5.3 *La fonction propre principale de l'opérateur linéaire correspondant à un noyau k défini par (5.4) est la fonction de norme* 1 *qui minimise l'erreur de reconstruction*

$$\int (k(x, y) - \lambda f(x) f(y))^2 p(x) p(y) dx dy. \tag{5.11}$$

Ce théorème est démontré dans (BENGIO, VINCENT et PAIEMENT 2003). On peut transformer l'équation (5.11) en un critère exempt de la condition de

norme unitaire en minimisant

$$\int (k(x,y) - g(x)g(y))^2 p(x)p(y)dxdy, \tag{5.12}$$

ce qui donne une solution g à partir de laquelle on peut retrouver λ et f en observant que $\lambda = ||g||^2$ et $f = g/\sqrt{\lambda}$.

La fonction f ainsi obtenue correspond à la projection des données observées sur la composante principale dans l'espace de caractéristiques, telle que défini par le théorème de Mercer.

Il est aussi possible de généraliser le théorème 5.3 à des fonctions propres arbitraires.

Théorème 5.4 *Étant données les $m-1$ fonctions propres principales f_i de l'opérateur linéaire associé à une fonction $k(x,y)$ comme à l'équation (5.4), la m-ième fonction propre peut être obtenue en minimisant l'espérance de*

$$\int \left(k(x,y) - g(x)g(y) - \sum_{i=1}^{m-1} \lambda_i f_i(x) f_i(y) \right)^2 p(x)p(y)dxdy \tag{5.13}$$

par rapport à g et à la densité $p(x,y) = p(x)p(y)$ ayant généré les données. On obtient alors la m-ième valeur propre principale $\lambda_m = ||g||^2$ et sa fonction propre correspondante $f_m = g/\sqrt{\lambda_m}$.

Ce théorème, démontré dans (Bengio, Vincent et Paiement 2003), suggère un algorithme par descente de gradient minimisant (5.13). Malheureusement, ce critère est de degré 4 par rapport à la fonction f, ce qui s'avère difficile à minimiser en pratique. Il a été impossible expérimentalement de minimiser ce critère par descente de gradient pour trouver la fonction propre principale d'un noyau gaussien en utilisant une classe non linéaire de fonctions implantée par des réseaux de neurones ou en utilisant une régression linéaire.

Une approche alternative est possible. En ne considérant que les fonctions de norme unitaire, la fonction f maximisant

$$\int f(x)k(x,y)f(y)p(x)p(y)dxdy \tag{5.14}$$

est la fonction propre principale de l'opérateur linéaire induit par k, donc minimisant (5.11). Le critère précédent est quadratique, positif semi défini et peut être maximisé par descente de gradient en pratique. La contrainte de norme unitaire est gérée par l'introduction d'un multiplicateur de Lagrange (BERTSEKAS 1995). Ainsi, le critère à optimiser devient

$$\max_f \min_\lambda \left[\int \int f(x)k(x,y)f(y)p(x)p(y)dxdy - \lambda \left(\int f(x)^2 p(x)dx - 1 \right) \right]. \tag{5.15}$$

Si la densité p est empirique, c'est-à-dire qu'on dispose d'un ensemble fini d'observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, le critère (5.15) devient alors

$$\max_f \min_\lambda \left[\frac{1}{n^2} \sum_{i=1}^{n} f(\boldsymbol{x}_i)k(\boldsymbol{x}_i, \boldsymbol{x}_j)f(\boldsymbol{x}_j) - \lambda \left(\frac{1}{n} \sum_{k=1}^{n} f(\boldsymbol{x}_k)^2 - 1 \right) \right]. \tag{5.16}$$

La première somme permet d'approximer les fonctions propres de k et la seconde contraint la norme de f. En pratique, il n'est pas nécessaire d'optimiser λ. En effet, le choix d'un multiplicateur de Lagrange arbitraire limite la norme de f à une constante finie. Il est alors possible de normaliser f en posant

$$\tilde{f} = \frac{f}{||f||}.$$

$\tilde{f}$ est alors la fonction propre principale associée à k, c'est-à-dire la fonction de norme unitaire maximisant (5.14) et minimisant (5.11).

Dans le cas où $f(\boldsymbol{x}) = \langle \mathbf{w}, \boldsymbol{x} \rangle$, la solution de (5.16) est donnée par le vecteur propre principal w_1 solution de l'équation de vecteurs propres généralisée

$$\frac{1}{n} X^t M X w_i = \lambda_i X^t X w_i.$$

La fonction propre principale a pu être estimée expérimentalement à l'aide d'un réseau de neurones maximisant le critère (5.16), pour le noyau associé à la segmentation spectrale. Cependant, des recherches sont toujours en cours afin d'établir un critère permettant d'estimer efficacement les fonctions propres subséquentes.

5.4 Expériences

On désire évaluer la précision de la généralisation des algorithmes à noyau élaborée aux sections 5.1 et 5.2.

On peut comparer cette précision à la perturbation intrinsèque des plongements si on substitue un certain nombre d'observation par d'autres tirées de la même distribution. On sépare les données en trois ensembles, $D = F \cup R_1 \cup R_2$ et on entraîne les algorithmes en utilisant $F \cup R_1$ ou bien $F \cup R_2$. On peut alors comparer les plongements obtenus sur F. Ainsi, la procédure du tableau 5.1 est appliquée à MDS, Isomap, LLE et la segmentation spectrale.

Le tableau 5.2 résume quels sont les noyaux généralisés présentés à la section 5.1 pour les différentes méthodes spectrales testées.

Les résultats de l'algorithme du tableau 5.1 pour MDS, Isomap, LLE et la segmentation spectrale sont présentés à la figure 5.2 pour différentes valeurs de $|R_1|/n$, donc de la proportion de points échangés (Bengio, Paiement et Vincent 2003). Les expérience sont réalisées sur une base de données de 698 images de visage synthétiques décrites par 4096 composantes. Cette base de données est disponible sur `http ://isomap.stanford.edu`. Les observations sont projetées sur les 2 composantes principales. Chaque expérience est répétée 5 fois en permutant les observations aléatoirement. On mesure la moyenne des résultats obtenus. Pour LLE, on reconstruit chaque point à l'aide des 10 plus proches voisins. Le même algorithme que pour Isomap est appliqué à MDS en connectant chaque point à tous les points de l'ensemble d'entraînement. Tous les algorithmes sont implantés grâce à la librairie C++ PLearn (`http ://www.plearn.org`) développée au laboratoire LISA (`http ://www.iro.umontreal.ca/~lisa`) de l'Université de Montréal.

Des résultats similaires ont été obtenus sur les bases de données Iosnosphere (`http ://www.ics.uci.edu/~mlearn/MLSummary.html`) et swissroll, (`www.cs.toronto.edu/~roweis/lle/`). Chaque algorithme génère un plongement en deux dimensions des images, de la même façon que les expériences réalisées avec Isomap dans (Tenenbaum, Silva et Langford 2000). Le nombre de voisins est de 10 pour Isomap et LLE et un noyau gaussien d'écart type 0.01 est utilisé pour la segmentation spectrale. Des intervalles de confiance

1. On choisit $F \subset D$ avec $m = |F|$ observations. Les $n-m$ observations restantes dans D/F sont séparées en deux sous-ensembles de même taille R_1 et R_2 ;
2. On entraîne chaque algorithme sur $F \cup R_1$ et $F \cup R_2$. Pour la segmentation spectrale, on applique l'algorithme présenté au tableau 4.5 ; Pour LLE, l'algorithme est présenté au tableau 4.3 ; Pour Isomap, on applique l'algorithme présenté au tableau 4.2 ; Finalement, on peut appliquer l'algorithme Isomap pour calculer MDS si on choisit le nombre de plus proches voisins k comme étant le nombre total de points de l'ensemble d'entraînement ;
3. Lorsque deux valeurs propres sont rapprochées, les vecteurs propres estimés ne sont pas stables et peuvent effectuer une rotation dans le sous-espace qu'ils génèrent. On estime un alignement affine entre les deux plongements en utilisant les projections des points de F en minimisant l'erreur quadratique de reconstruction des points. On calcule par la suite la distance euclidienne entre les points des plongements alignés obtenus pour chaque $\boldsymbol{x}_i \in F$;
4. Pour chaque observation $\boldsymbol{x}_i \in F$, on entraîne aussi chaque algorithme sur $\{F \cup R_1\}/\{\boldsymbol{x}_i\}$ de la même façon qu'à l'étape 2 ;
5. On applique la formule (5.5) (Nyström) aux points hors échantillons pour trouver la prédiction du plongement de $\boldsymbol{x}_i$. On utilise pour ce faire les noyaux généralisés résumés au tableau 5.2 en fonction de l'algorithme testé ;
6. On calcule la distance entre la prédiction obtenue à l'étape 5 et le plongement obtenu avec le même algorithme entraîné sur $F \cup R_1$ à l'étape 2, qui contient donc $\boldsymbol{x}_i$ dans son ensemble d'entraînement ;
7. On calcule la différence moyenne et son écart type entre la distance obtenue à l'étape 3 et celle obtenue à l'étape 6 pour chaque $\boldsymbol{x}_i \in F$. On répète cette expérience pour différentes cardinalités de F.

Tableau 5.1 – *Comparaison empirique entre la variation intrinsèque des algorithmes et l'erreur de généralisation.*

Algorithme	Noyau généralisé $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
Segmentation spectrale	En utilisant les notations du tableau 4.5, le noyau généralisé est trivialement donné par $$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{n\tilde{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{\sum_{k=1}^{m} \tilde{K}(\boldsymbol{x}_k, \boldsymbol{x}_i) \sum_{l=1}^{m} \tilde{K}(\boldsymbol{x}_j, \boldsymbol{x}_l)}}$$ où $\tilde{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma^2)$.
Isomap	On utilise les notations du tableau 4.2. Ajouter les points $\boldsymbol{x}_i$ et $\boldsymbol{x}_j$ au graphe G si ils y sont absents. Connecter ces points aux k plus proches voisins dans G. $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ = plus court chemin entre $\boldsymbol{x}_i$ et $\boldsymbol{x}_j$ dans G.
LLE	Tel que défini à la section 5.1.4, on pose $$k(\boldsymbol{x}, \boldsymbol{y}) = w(\boldsymbol{x}, \boldsymbol{y}) + w(\boldsymbol{y}, \boldsymbol{x}) - \sum_{i=1}^{m} w(\boldsymbol{x}_i, \boldsymbol{x}) w(\boldsymbol{x}_i, \boldsymbol{y}).$$ où $w(\boldsymbol{x}, \boldsymbol{y})$ est défini en (5.3).
MDS	$k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ = même chose que pour Isomap en connectant $\boldsymbol{x}_i$ et $\boldsymbol{x}_j$ à tous les points de l'ensemble d'entraînement.

Tableau 5.2 – *Noyaux généralisés pour les différentes méthodes spectrales testées*

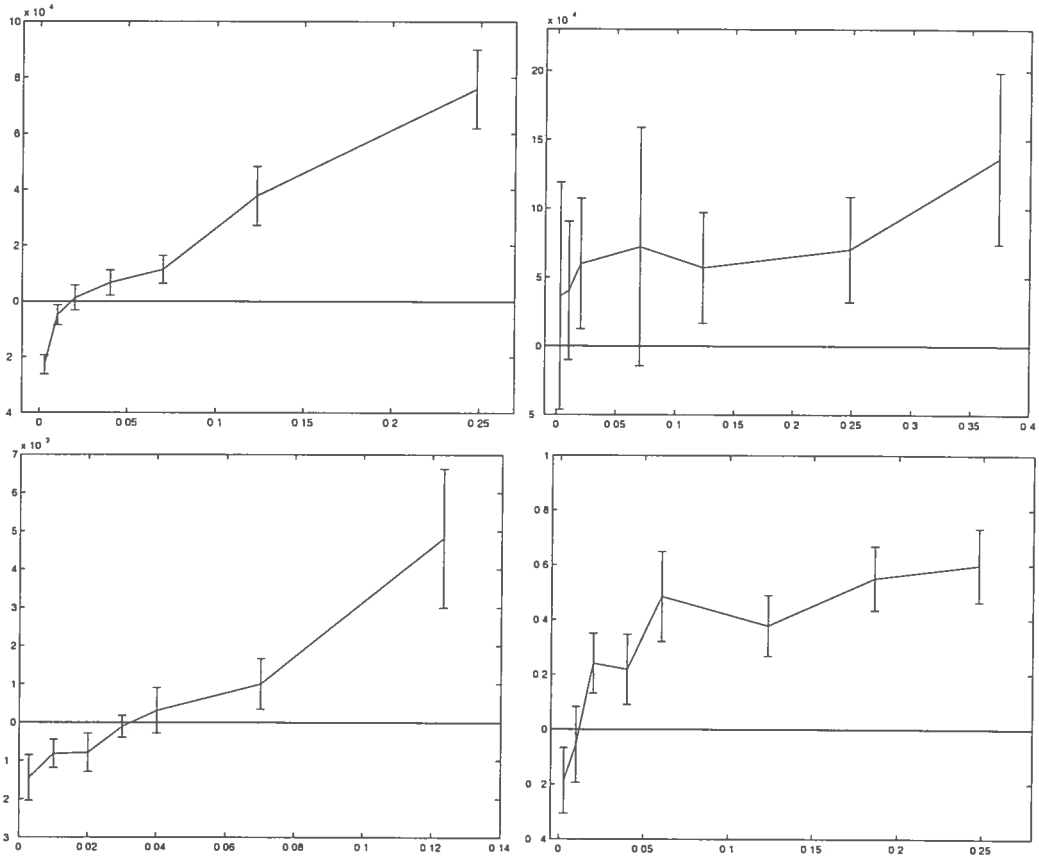

Figure 5.2 – Variabilité sur l'ensemble d'entraînement moins l'erreur hors échantillon par rapport à la proportion d'exemples d'entraînement substitués. En haut à gauche : MDS. En haut à droite : segmentation spectrale. En bas à gauche : Isomap. En bas à droite : LLE. Les intervalles de confiance sont de 95%. L'utilisation de la formule de Nyström correspond à changer environ 2% des exemples de l'ensemble d'entraînement.

de 95% sont illustrés autour de chaque moyenne de différences d'erreurs sur la figure.

Comme l'intuition le suggère, la différence moyenne entre les deux distances augmente de façon presque monotone par rapport à l'augmentation du nombre $|R_1|$ d'observations d'entraînement. Ce comportement est attribuable à l'augmentation de la variabilité du plongement obtenu par les algorithmes lorsqu'un nombre croissant d'observations sont substituées.

On constate que l'erreur hors échantillon est moins importante que la variabilité des algorithmes lorsque plus d'environ 2% des exemples d'entraînement sont substitués pour les algorithmes MDS et LLE et lorsque plus d'environ 3% de ces exemples sont substitués pour l'algorithme Isomap. Pour la segmentation spectrale, la variabilité des algorithmes est toujours supérieure à l'erreur de généralisation hors échantillon. Cette meilleure performance pour la généralisation de la segmentation spectrale est probablement attribuable au fait que le noyau généralisé utilisé dans ce cas est en fait le noyau original utilisé aussi sur les observations de l'ensemble d'entraînement.

Les courbes associées aux algorithmes MDS et Isomap augmentent de façon monotone avec l'augmentation du nombre d'exemples substitués aléatoirement, ce qui suggère un comportement plus stable de ces algorithmes. D'autre part, les algorithmes de segmentation spectrale et LLE semblent dans ce cas plus sensibles aux variations de l'ensemble d'entraînement.

CHAPITRE 6

Conclusion

Les algorithmes Isomap, LLE, et la segmentation spectrale sont des méthodes de réduction de dimension ou de segmentation s'affranchissant des limites inhérentes aux approches linéaires. Cependant, ces algorithmes n'offraient pas de méthode directe de généralisation efficace à des exemples hors échantillon sans calculer à nouveau les vecteur propres des matrices de Gram correspondantes. On montre donc que la formule de Nyström peut être utilisée dans cette perspective. On a constaté que ces algorithmes réalisent en fait la même tâche définie sur des noyaux différents. En effet, on estime les fonctions propres d'un opérateur linéaire associé à un noyau et à la distribution sous-jacente des données. On a aussi montré que ces méthodes minimisent une fonction de coût définissant la qualité d'une généralisation. La formule de Nyström est une extension possible, mais on pourrait concevoir d'autres estimateur des fonctions propres de l'opérateur linéaire G. Lorsque les noyaux sont positifs semi définis, les algorithmes présentés peuvent être vu comme une application de l'ACP à noyau en choisissant un noyau particulier. On note que Isomap induit généralement un noyau possédant des valeurs propres négatives, ce qui justifie la conception d'une théorie où les noyaux ne doivent pas nécessairement être positifs semi définis.

Des expériences ont montré empiriquement sur plusieurs bases de données que les plongements hors échantillon prédits sont en général assez près de ceux qui auraient été obtenus en incluant les points de test dans l'ensemble d'entraînement. Les distances obtenues sont en effet comparables aux perturbations obtenues par de petites modifications de l'ensemble d'entraînement.

La principale contribution de ce mémoire est la définition d'une fonction de coût pour les méthodes spectrales présentées permettant de généraliser ces algorithmes à des points hors échantillon avec un coût proportionnel au nombre d'observations dans l'ensemble d'entraînement. Cette approche est une extension directe du travail entrepris par (WILLIAMS et SEEGER 2000), où on constate que la formule de Nyström approxime les fonctions propres des opérateurs linéaires associés aux noyaux des méthodes spectrales. Des travaux sur la convergence de l'erreur de généralisation de l'ACP à noyau (SHAWE-TAYLOR, CRISTIANINI et KANDOLA 2002; SHAWE-TAYLOR et WILLIAMS 2003) aident aussi à justifier le fait que les extensions proposées estiment la limite convergente des vecteurs propres de matrices de Gram, lorsque les noyaux sont semi définis positifs. (HAM, LEE et SCHÖLKOPF 2003) ont déjà proposé l'introduction des noyaux dans la définition de LLE et Isomap, mais n'ont pas proposé d'extension de ces algorithmes aux points hors échantillon.

6.1 Pistes de recherche

Plusieurs questions demeurent ouvertes et pourraient donner lieu à la conception d'algorithmes de réduction de dimension plus souples et performants. Le problème majeur affligeant les méthodes spectrales présentées est la nécessité de disposer d'ensemble de données contenant suffisament d'observations pour bien caractériser la courbure des variétés sur lesquelles sont situées ces données. On constate bien souvent en pratique que le nombre d'observations disponibles situées sur des régions très courbées des variétés à estimer est souvent nettement insuffisant pour obtenir des résultats satisfaisants. Bien entendu, la façon de caractériser quantitativement les observations joue

un rôle capital dans la forme que prennent les variétés dans l'espace d'observation. Le fait de caractériser les observations par des valeurs induisant des variétés de faible courbure pourrait sans doute accroître l'efficacité des méthodes spectrales appliquées à ces observations.

On pourrait utiliser une distribution plus lisse que la distribution empirique pour définir l'opérateur linéaire G_n. Intuitivement, une distribution qui serait plus proche de la vraie distribution ayant généré les données aurait de meilleures chances de produire une bonne généralisation en estimant mieux les fonctions propres de G.

Dans un autre ordre d'idées, les algorithmes spectraux présentés n'offrent pas de méthode directe pour estimer la position d'une observation dans l'espace d'observation à partir de ses coordonnées dans l'espace de dimension réduite. Si le nombre de points est suffisant, on pourrait appliquer une méthode similaire au positionnement multidimensionnel afin de respecter les distances dans l'espace d'observation. De fait, les méthodes spectrales pourraient alors devenir des algorithmes de compression dans des applications où la qualité de la compression revêt plus d'importance que la vitesse de compression.

D'autre part, toutes les méthodes présentées capturent les caractéristiques saillantes de la densité inconnue ayant généré les données. Il serait peut-être possible d'utiliser les représentations apprises à travers les fonctions propres estimées pour construire un estimateur de densité dans l'espace d'observation. La modélisation des points dans l'espace de dimension réduite pourrait ainsi s'avérer beaucoup plus simple que dans l'espace d'observation de haute dimension.

La pertinence des vecteurs propres associés à des valeurs propres négatives d'une matrice de Gram est mal comprise. Bien que le théorème de Mercer ne puisse s'appliquer à des noyaux qui ne sont pas semi définis positifs, on constate empiriquement que les vecteurs propres associés à des valeurs propres négatives capturent de l'information complémentaire à celle capturée par les vecteurs propres de valeur propre positive. Le noyau associé à l'algorithme Isomap n'est pas positif semi défini et sa matrice de Gram a donc des valeur propres négatives. L'algorithme Isomap ne fait qu'ignorer les vecteurs

propres associés à ces valeurs propres. Des méthodes tenant compte de ces vecteurs propres pourraient produire des plongement de plus faible dimension préservant d'avantage d'information sur les observations d'origine.

Le taux de convergence des fonctions propres trouvées à l'aide de la formule de Nyström n'a pas été mesuré empiriquement et on ne connaît pas de valeurs théoriques bornant ce taux. Les résultats empiriques positifs obtenus à la section 5.4 indiquent toutefois que ce taux est acceptable en pratique dans les contextes testés.

Des variantes sont possibles pour les noyaux généralisés définis à la section 5.1. Certaines de ces variantes pourraient s'avérer plus précises ou efficaces en pratique. De plus, il serait peut-être possible de définir des noyaux généralisés satisfaisant les conditions de convergence du théorème 5.2. Inversement, les conditions d'applicabilité de ce théorème pourraient être allégées de façon à pouvoir y inclure des noyaux plus généraux. On constate empiriquement que les noyaux définis à la section 5.1 sont convergents même s'ils ne satisfont pas aux conditions du théorème.

Des critères convexes pourraient être définis afin de pouvoir estimer avec des classes de fonctions assez vastes comme les réseaux de neurones les fonctions propres principales de G, de façon a obtenir des estimateurs de ces fonctions pouvant mieux généraliser les plongements spectraux qu'avec la formule de Nyström.

Finalement, des algorithmes pourraient apprendre des concepts à des niveaux d'abstraction élevés en se basant sur des abstraction de niveau plus bas. Les algorithmes non supervisés peuvent être appliqués en couches multiples. Des structures moins locales pourraient alors être apprises en se basant sur des représentations abstraites des données.

6.2 Applicabilité des résultats obtenus

Le fait de généraliser les algorithmes spectraux aux observations hors échantillon élargit considérablement leur champs d'application. En effet, le

coût d'entraînement quadratique de ces algorithmes est moins problématique en pratique, dans la mesure où l'entraînement des algorithmes peut être réalisé avant leur application à un problème à résoudre pour lequel le temps d'exécution acceptable est limité. La formule de Nyström offre pour cela une projection des point hors échantillon à un coût linéairement proportionnel au nombre d'exemples de l'ensemble d'entraînement.

Les algorithmes de réduction de dimension linéaires sont déjà utilisés depuis bon nombre d'années en forage de données pour des applications de marketing, de détection de fraude ou d'actuariat. La grande majorité des ensembles de données utilisés dans ces applications contiennent des variables aléatoires reliées par des dépendances non linéaires. Il est donc tout a fait légitime de penser que l'utilisation de méthodes spectrales non linéaires pourra améliorer de façon sensible la précision des systèmes utilisés dans de telles applications.

On constate aussi l'utilisation accrue depuis quelques années des méthodes spectrales dans des domaines aussi variés que la vision artificielle, les biotechnologies, le traitement des langues naturelles ou la finance, pour n'en nommer que quelques-uns. En fait, l'application de tels algorithmes peut se faire naturellement dans n'importe quel domaine où interviennent de grandes quantités de données situées dans un espace de haute dimension.

Finalement, l'aspect le plus attrayant des méthodes présentées et de celles qui pourraient en découler réside dans l'interprétabilité des résultats obtenus. En effet, appliquer de façon fructueuse un algorithme de réduction de dimension à un problème où des données sont exprimées par un grand nombre de caractéristiques fortement dépendantes revient à isoler les mécanismes fondamentaux régissant les variations observées dans cet ensemble de données. La résolution de ce problème constitue sans conteste l'un des objectifs fondamentaux de la recherche scientifique en général.

Références

ARONSZAJN, N. (1950), « Theory of reproducing kernels », *Transactions of the American Mathematical Society 68*, p. 337–404.

BAKER, C. (1977), *The numerical treatment of integral equations*, Oxford : Clarendon Press.

BENGIO, Y., O. DELALLEAU, N. LEROUX, J.-F. PAIEMENT, P. VINCENT et M. OUIMET (2003), « Spectral Clustering and Kernel PCA are Learning Eigenfunctions », Rapport technique 1239, Université de Montréal.

BENGIO, Y., J.-F. PAIEMENT et P. VINCENT (2003), « Out-of-sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering », *Advances in Neural Information Processing Systems*, The MIT Press,

BENGIO, Y., P. VINCENT et J.-F. PAIEMENT (2003), « Learning Eigenfunctions of Similarity : Linking Spectral Clustering and Kernel PCA », Rapport technique 1232, Université de Montréal, Soumis à Neural Computation.

BERG, C., J. CHRISTENSEN et P. RESSEL (1984), *Harmonic Analysis on Semigroups*, New York : Springer-Verlag.

BERTSEKAS, D. P. (1995), *Nonlinear Programming.* Athena Scientific.

BILLINGSLEY, P. (1995), *Probability and Measure*, New York : John Wiley and Sons.

BISHOP, C. M. (1995), *Neural Networks for Pattern Recognition.* Oxford University Press.

BISHOP, R. et S. GOLDBERG (1980), *Tensor Analysis on Manifolds*, New

York : Dover.

BREIMAN, L. (1968), *Probability*, Reading, MA : Addison-Wesley.

CHAVEL, I. (1993), *Riemannian geometry : a modern introduction.* Cambridge University Press.

COX, T. et M. COX (1994), *Multidimensional Scaling*, London : Chapman & Hall.

CRISTIANINI, N. et J. SHAWE-TAYLOR (2000), *An introduction to support vector machines.* Cambridge University Press.

DIAMANTARAS, K. et S. KUNG (1996), *Principal Components Neural Networks.* Wiley-Interscience.

FELLER, W. (1971), *An Introduction to Probability Theory and its Applications*, New York : John Wiley and Sons, 2nd edition.

GOWER, J. (1968), « Adding a point to vector diagrams in multivariate analysis », *Biometrika 55*(3), p. 582–585.

GOWER, J. C. (1966), « Some distance properties of latent root and vector methods in multivariate analysis », *Biometrika 53*, p. 315–328.

HAM, J., D. LEE et B. SCHÖLKOPF (2003), « A kernel view of the dimensionality reduction of manifolds », Rapport technique TR-110, Max Planck Institue for Biological Cybernetics, Germany.

HINTON, G. E. et T. J. SEJNOWSKI (Édits.) (1999), *Unsupervised Learning and Map Formation : Foundations of Neural Computation*, Cambridge, MA : MIT Press.

HORN, R. A. et C. R. JOHNSON (1990), *Matrix Analysis*, Cambridge : Cambridge University Press.

HOTELLING, H. (1933), « Analysis of a Complex of Statistical Variables into Principal Components », *J. Educ. Psychol. 24*, p. 339–354.

KÖNIG, H. (1986), *Eigenvalue Distribution of Compact Operators*, Basel : Birkhäuser.

KOLMOGOROV, A. N. et S. V. FOMIN (1961), *Functional Analysis*, Albany, NY : Graylock Press.

LANG, S. (1989), *Linear Algebra*, New York : Springer-Verlag.

LEE, J. M. (1997), *Riemannian Manifolds : An Introduction to Curvature.* Springer-Verlag.

LITTMAN, M. L., D. F. SWAYNE, N. DEAN et A. BUJA (1992), « Computing Science and Statistics », *Proceedings of the 24th Symposium on the Interface*, Fairfax Station, VA, p. 208–217.

MALIK, J., S. BELONGIE, T. LEUNG et J. SHI (2000), « Contour and texture analysis for image segmentation », *Perceptual Organization for Artificial Vision Systems.* Kluwer.

MARDIA, K. V. (1978), « Some properties of classical multidimensional scaling », *Comm. Statist.-Theor. Meth 7*, p. 1233–1241.

MARDIA, K. V., J. T. KENT et J. M. BIBBY (1979), *Multivariate Analysis*, Probability and Mathematical Statistics. London : Academic Press.

MERCER, J. (1909), « Functions of positive and negative type and their connection with the theory of integral equations », *Philosophical Transactions of the Royal Society, London 209*, p. 415–446.

MUNKRES, J. R. (1975), *Topology : A First Course*, Englewood Cliffs, NJ : Prentice-Hall.

NG, A. Y., M. I. JORDAN et Y. WEISS (2002), « On spectral clustering : Analysis and an algorithm », *Advances in Neural Information Processing Systems*, Cambridge, MA, MIT Press,

PEARSON, K. (1901), « On Lines and Planes of Closest Fit to Systems of Points in Space », *Philos. Mag., Ser. 6 2*, p. 559–572.

REED, M. et B. SIMON (1980), *Methods of modern mathematical physics. Vol. 1 : Functional Analysis*, San Diego : Academic Press.

ROWEIS, S. T. et L. K. SAUL (2000, December), « Nonlinear Dimensionality Reduction by Locally Linear Embedding », *Science 290*(5500), p. 2323–2326.

RUDIN, W. (1995), *Principes d'analyse mathématique.* Ediscience international.

SAITOH, S. (1988), *Theory of reproducing kernels and its applications*, Harlow, England : Longman Scientific & Technical.

SAUL, L. K. et S. T. ROWEIS (2003), « Think Globally, Fit Locally : Unsupervised Learning of Low Dimensional Manifolds », *Journal of Machine Learning Research 4*, p. 119–155.

SCHÖLKOPF, B., A. SMOLA et K.-R. MÜLLER (1998), *Advances in Kernel Methods*, Chapter 20, p. 327–352. MIT Press.

SCHÖLKOPF, B. (1997), *Support Vector Learning*, Ph. D. thesis, Technische Universität Berlin, R. Oldenburg Verlag, München, Doktorarbeit.

SCHÖLKOPF, B. et A. SMOLA (2002), *Learning with Kernels*. MIT Press.

SCHÖLKOPF, B., A. SMOLA et K.-R. MÜLLER (1998), « Nonlinear component analysis as a kernel eigenvalue problem », *Neural Computation 10*, p. 1299–1319.

SCHOENBERG, I. J. (1935), « Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espaces distanciés vectoriellement applicable sur l'espace de Hilbert" », *Ann. Math. 36*, p. 724–732.

SHAWE-TAYLOR, J., N. CRISTIANINI et J. KANDOLA (2002), « On the concentration of spectral properties », *Advances in Neural Information Processing Systems*, The MIT Press,

SHAWE-TAYLOR, J. et C. WILLIAMS (2003), « The stability of kernel principal components analysis and its relation to the process eigenspectrum », *Advances in Neural Information Processing Systems*, The MIT Press,

TENENBAUM, J. B., V. D. SILVA et J. C. LANGFORD (2000), « A Global Geometric Framework for Nonlinear Dimensionality Reduction », *Science 290*(5500), p. 2319–2323.

TORGERSON, W. S. (1958), *Theory and Methods of Scaling*, New York : Wiley.

VAPNIK, V. (1995), *The Nature of Statistical Learning Theory*, New York : Springer-Verlag.

VAPNIK, V. (1998), *Statistical Learning Theory*, New York : Wiley.

WAHBA, G. (1990), « Spline models for observational data », *CBMS-NSF Regional Conference Series in Applied Mathematics*, Philadelphia, PA,

WEISS, Y. (1999), « Segmentation using eigenvectors : unifying view », *Proceedings of the IEEE International Conference on Computer Vision*, p. 975–982.

WILLIAMS, C. (2001), « On a connection between kernel pca and metric multidimensional scaling », *Advances in Neural Information Processing Systems*, The MIT Press, p. 675–681.

WILLIAMS, C. et M. SEEGER (2000), « The effect of the input density distribution on kernel-based classifiers », *Proceedings of the Seventeeth International Conference on Machine Learning*, Morgan Kaufmann,

YOUNG, G. et A. S. HOUSEHOLDER (1938), « Discussion of a set of points in terms of their mutual distances », *Psychometrika 3*, p. 19–22.