
PROJET 1

 LIEN [GITHUB](#)

JEAN-THOMAS BAILLARGEON
CHRISTOPHER BLIER-WONG
POUR LE COURS STT-7330
MÉTHODE D'ANALYSE DES DONNÉES

PRÉSENTÉ LE 25 MARS 2018 À LA PROFESSEURE

ANNE-SOPHIE CHAREST

*Département de mathématiques et de statistiques
Faculté des sciences et de génie
Université Laval*



JEAN-THOMAS BAILLARGEON
CHRISTOPHER BLIER-WONG
FACULTÉ DES SCIENCES ET DE GÉNIE
ÉCOLE D'ACTUARIAT
UNIVERSITÉ LAVAL
HIVER 2018

Table des matières

1	Introduction	2
2	Motivation	2
3	Fonction noyau	4
4	PCA avec noyau	5
4.1	Normalisation des données	6
5	Application pratique	7
6	Discussion	9
6.1	Avantages	9
6.2	Désavantages	10
7	Conclusion	11

1 Introduction

Dans la résolution de problèmes reliés à l'étude des données, le statisticien peut être confronté à des jeux de données volumineux ayant un grand nombre d'attributs. Ce genre de jeux de données posent certaines difficultés telles un temps d'exécution élevé, le fléau de la dimension ou encore le sur-apprentissage. Afin de palier à ces problèmes, quelques méthodes efficaces et reconnues ont été développées afin de réduire mathématiquement le nombre de variables explicatives d'un jeu de données. Une de ces techniques est l'analyse par composantes principales (ACP). Ce rapport présente une généralisation de la méthode permettant d'y intégrer des éléments de non linéarité. La motivation de cette généralisation est de permettre de capturer autant de variance tout en réduisant davantage le nombre de composantes principales (CP) utilisées.

2 Motivation

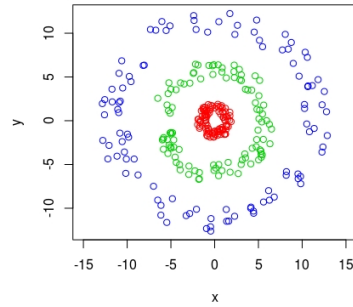
L'analyse par composantes principales est une technique de réduction de la dimension d'un jeu de données. Elle permet de projeter les données dans un espace restreint tout en conservant le maximum de variance entre les données. Les axes utilisés par l'analyse sont appelées composantes principales. Ces composantes principales permettent de transposer les données en utilisant les meilleures combinaisons linéaires. Les composantes ordonnées de telle sorte qu'elles conservent une proportion décroissante de la variance du jeu de données. Les premières composantes sont généralement beaucoup plus informatives que les dernières. Il est ainsi possible de réduire la dimension d'un jeu de données en ne considérant que les premières composantes principales tout en conservant une quantité d'information satisfaisante.

Mathématiquement, l'ACP s'exécute en décomposant la matrice de variance-covariance (ou de corrélation) estimée avec le jeu de données en valeurs et vecteurs propres. La matrice de variance-covariance est définie telle que

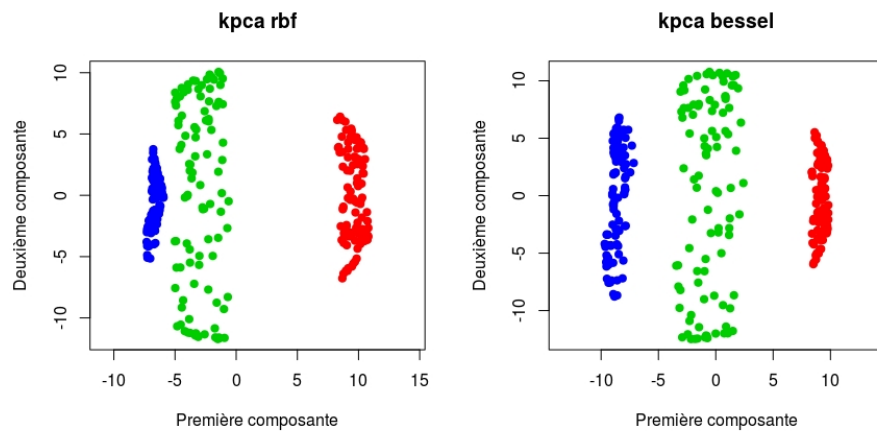
$$C = Var(X) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \cdot \mathbf{x}_i$$

La principale limitation de cette méthodologie est qu'elle ne permet pas d'exploiter l'information contenue dans les interactions non linéaires entre les attributs. De plus, vu que les composantes principales sont indépendantes les unes entres elles, l'information non linéaire peut réapparaître dans les données en conservant un grand nombre de composantes principales.

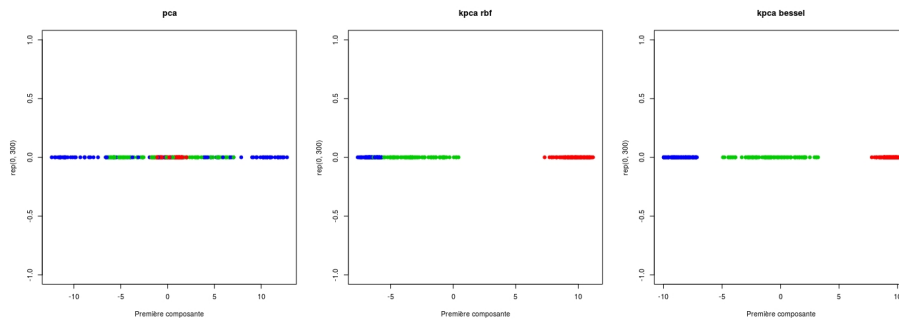
Par exemple, avec les données dans la figure suivante n'ont aucune de relation linéaire.



Ensuite, on applique deux exemples de transformations non linéaires sur les données et on obtient



On remarque que ces données peuvent être séparés et interprétés linéairement. On peut donc appliquer une ACP sur ces transformations et on obtient



L'idée de l'espace des attributs est de projeter les données originales par une fonction non linéaire vers un nouvel espace qui permet d'interpréter linéairement les relations entre les attributs. Formellement, on a

$$\begin{aligned}\Phi : \mathbb{R}^N &\rightarrow \mathcal{F} \\ \mathbf{x} &\mapsto \Phi(\mathbf{x})\end{aligned}\tag{2.1}$$

où les données $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \in \mathbb{R}^p$ est projeté vers un espace d'attributs \mathcal{F} [Muller et al., 2001]. Souvent, la dimension de \mathcal{F} est beaucoup plus élevée que p . L'apprentissage statistique peut maintenant être fait sur les données

$$(\Phi(\mathbf{x}_1), y_1), (\Phi(\mathbf{x}_2), y_2), \dots, (\Phi(\mathbf{x}_n), y_n).$$

Afin de motiver cette généralisation du modèle, considérons le jeu de donnée avec $p = 2$, où

$$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N = (x_{1,1}, x_{1,2}), (x_{2,1}, x_{2,2}), \dots, (x_{N,1}, x_{N,2})$$

et la transformation $\Phi(\mathbf{x}) = (x_1^2 + \sqrt{2} \times x_1 \times x_2 + x_2^2)$. Visuellement, on projète les données de cercles vers une demi-sphère, où les données dans le centre cercle ont une profondeur faible et les données aux périmètre du cercle ont une profondeur élevée. De plus, la matrice de variance covariance à utiliser pour l'ACP devient

$$\overline{C} = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T.$$

Ensuite, on peut reformuler le produit scalaire entre les deux espaces d'attributs comme

$$\begin{aligned}(\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) &= (x_{i,1}^2 + \sqrt{2}x_{i,1}x_{i,2} + x_{i,2}^2)(x_{j,1}^2 + \sqrt{2}x_{j,1}x_{j,2} + x_{j,2}^2)^T \\ &= ((x_{i,1}x_{i,2})(x_{j,1}x_{j,2})^T)^2 \\ &= (\mathbf{x}_i \cdot \mathbf{x}_j)^2.\end{aligned}$$

En remplaçant avec le résultat précédent, on obtient

$$\overline{C} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i \cdot \mathbf{x}_j)^2$$

et on doit appliquer une décomposition par valeurs et vecteurs propres sur cette matrice pour effectuer une ACP avec noyau, où le noyau est $(\mathbf{x}_i \cdot \mathbf{x}_j)^2$.

3 Fonction noyau

On rappelle la propriété suivante du produit scalaire pour les vecteurs \mathbf{x} et \mathbf{y} :

$$\mathbf{x}\mathbf{y}^T = \mathbf{x} \cdot \mathbf{y}.$$

En général, lorsque le produit scalaire entre les transformations définies en (2.1) existe, on appelle cette fonction le noyau et on peut reformuler le produit scalaire comme

$$k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x})\Phi(\mathbf{y})^T = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}).$$

Il est important de rappeler que pour obtenir l'information k générée par la projection, les transformations \mathcal{F} n'ont jamais été explicitement calculés. La fonction de noyau permet de calculer directement cette information et il s'agit de la raison pour laquelle les noyaux sont des outils intéressants. Dans plusieurs problèmes d'apprentissage, le "truc du noyau" permet d'éviter de calculer directement les nouvelles données $\Phi(\mathbf{x})$. En effet, il n'est parfois pas nécessaire d'avoir toutes les données car on peut reformuler les équations de mise à jour par le produit scalaire entre différents $\Phi(\mathbf{x})$ et ainsi les remplacer par la fonction de noyau.

L'exemple jouet ne montrait qu'une projection en 3 dimensions. Cependant, certains noyaux émulent une projection dans un espace de dimensions infinie. En projetant dans un espace ayant un nombre de dimensions infini, il est théoriquement possible de modéliser parfaitement chacun des points des données. On a ainsi l'assurance de trouver un hyperplan séparant les données parfaitement - sans avoir à faire des calculs qui seraient impossibles à faire.

En pratique, au lieu de sélectionner un espace d'attributs \mathcal{F} , on sélectionne directement le noyau. Il n'est pas nécessaire de savoir l'espace d'attributs \mathcal{F} ou la fonction Φ . De plus, il est possible que la fonction Φ projette des données vers une dimension infinie mais cette fonction n'a pas besoin d'être connue car l'impossibilité analytique de déterminer cette fonction est esquivée par le truc du noyau. Des exemples de noyaux communément utilisés sont présentés dans la table 1.

Nom	$k(\mathbf{x}, \mathbf{y})$
Gaussien (RBF)	$\exp\left(\frac{-\ \mathbf{x}-\mathbf{y}\ ^2}{c}\right)$
Polynomial	$((\mathbf{x} \cdot \mathbf{y} + \theta))^d$
Sigmodoidal	$\tanh(\kappa(\mathbf{x} \cdot \mathbf{y}) + \theta)$
Multiquadrique inversé	$\frac{1}{\sqrt{\ \mathbf{x}-\mathbf{y}\ ^2 + c^2}}$

TABLE 1 – Noyaux communs

4 PCA avec noyau

Dans le contexte de l'ACP, on trouvait les valeurs et vecteurs propres qui correspondaient à la matrice de covariance. Si les données sont centrées, i.e. $\sum_{i=1}^N x_i = 0$, on a

$$C = \text{Var}(X) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T.$$

[Schölkopf et al., 1997] proposent de répéter cette analyse sur une transformation des données originales, i.e. trouver les valeurs et vecteurs propres de

$$\overline{C} = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T. \quad (4.1)$$

Ensuite, en appliquant le "truc du noyau" présenté dans la section (3), on évite de calculer explicitement les données $\Phi(\mathbf{x})$, il suffit de calculer la matrice des noyaux. Comme dans l'ACP, on doit faire une décomposition par valeurs et vecteurs propres de

$$\overline{C} = \frac{1}{N} \sum_{i=1}^N K(\mathbf{x}_i, \mathbf{x}_i). \quad (4.2)$$

4.1 Normalisation des données

Dans le développement de l'ACP, on a appliqué l'hypothèse que les données étaient centrées. Par contre, dans le contexte de projection des données dans l'espace \mathcal{F} et pour profiter du truc du noyau, on ne peut pas calculer

$$\tilde{\Phi}(\mathbf{x}_i) = \Phi(\mathbf{x}_i) - \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i).$$

La solution à ce problème est présentée dans [Schölkopf et al., 1998]. Soit la matrice K , où

$$K_{ij} = (k(\mathbf{x}_i, \mathbf{x}_j))_{ij}.$$

Il est possible que $\frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}) \neq 0$. On doit alors centrer les données selon le développement suivant :

$$\begin{aligned} \tilde{k}(\mathbf{x}_i, \mathbf{x}_j) &= \tilde{\Phi}(\mathbf{x}_i)^T \tilde{\Phi}(\mathbf{x}_j) \\ &= \left(\Phi(\mathbf{x}_i) - \frac{1}{N} \sum_{l=1}^N \Phi(\mathbf{x}_l) \right)^T \left(\Phi(\mathbf{x}_j) - \frac{1}{N} \sum_{l=1}^N \Phi(\mathbf{x}_l) \right) \\ &= k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{N} \sum_{l=1}^N k(\mathbf{x}_i, \mathbf{x}_l) - \frac{1}{N} \sum_{l=1}^N k(\mathbf{x}_j, \mathbf{x}_l) + \frac{1}{N^2} \sum_{l,k} k(\mathbf{x}_l, \mathbf{x}_k). \end{aligned}$$

Alors, on peut centrer la matrice \tilde{K} selon

$$\tilde{K}_{ij} = K_{ij} - \mathbb{1}_N K - K \mathbb{1}_N + \mathbb{1}_N K \mathbb{1}_N,$$

où $(\mathbb{1}_N)_{ij} := \frac{1}{N}$ et éviter de calculer les données $\Phi(\mathbf{x})$.

5 Application pratique

On applique l'analyse par composantes principales et l'analyse par composantes principales avec noyau sur le jeu de données MNIST. Une donnée représente l'intensité de gris entre 0 et 255 des 728 pixels d'un image 28×28 de chiffres entre 0 et 9. Elles ont été récoltées par [LeCun et al., 1998]. Voici un exemple des données :

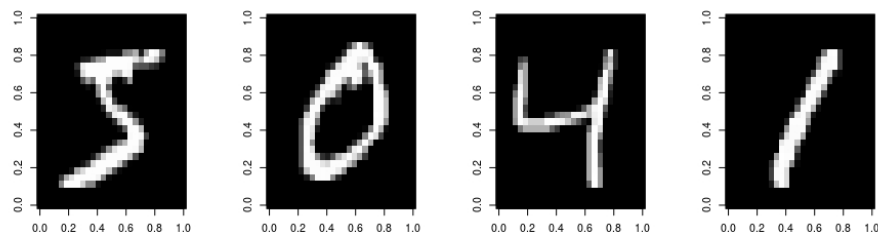


FIGURE 1 – 4 premiers exemples de MNIST

On crée aussi un jeu de données modifié de MNIST, où on applique un bruit gaussien. Voici les mêmes exemples que dans la figure 1 :

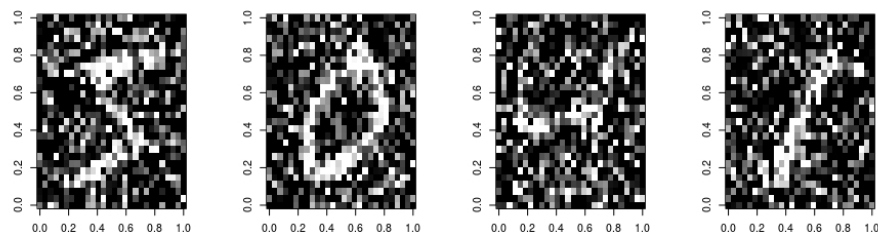
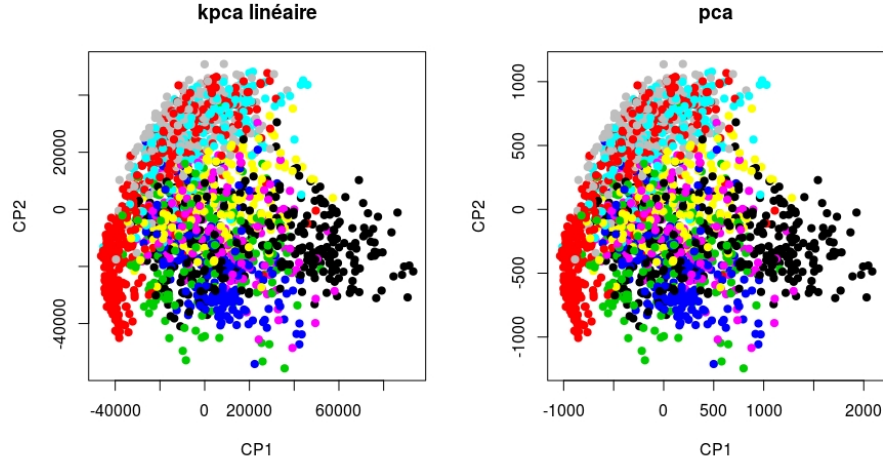


FIGURE 2 – 4 premiers exemples de MNIST avec bruit gaussien

Une analyse par composantes principales est identique à une analyse par composantes principales avec un noyau linéaire. On projette les données selon les deux premières composantes principales. On obtient



Il n'y a pas de flexibilité à cette méthode. On remplace la matrice variance-covariance par différents noyaux et on présente les projections dans la prochaine figure.

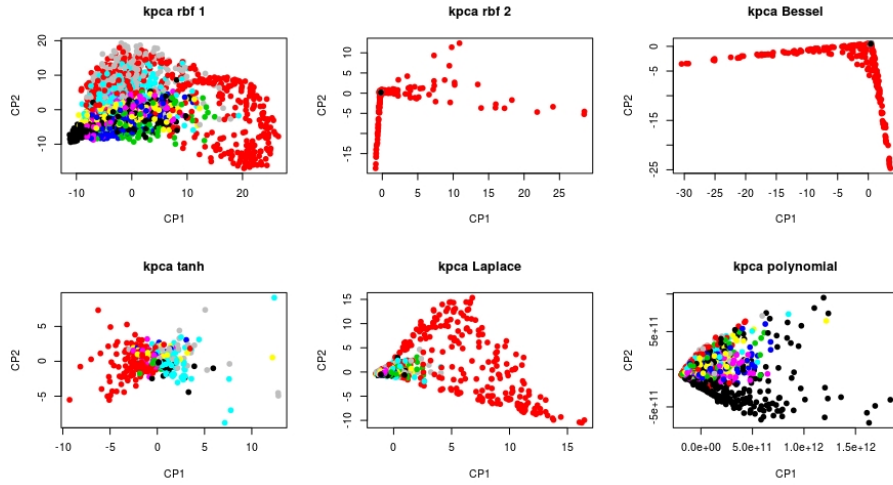


FIGURE 3 – Projection des deux premières composantes principales selon différents noyaux.

On remarque que le choix du noyau a beaucoup d'importance sur la projection. Les données en rouge représentent le chiffre 1 et la plupart des noyaux peuvent séparer les données. Le noyau polynomial est performant pour segmenter les chiffres 0.

On applique ensuite l'ACP avec noyau sur les données MNIST bruitées. On obtient

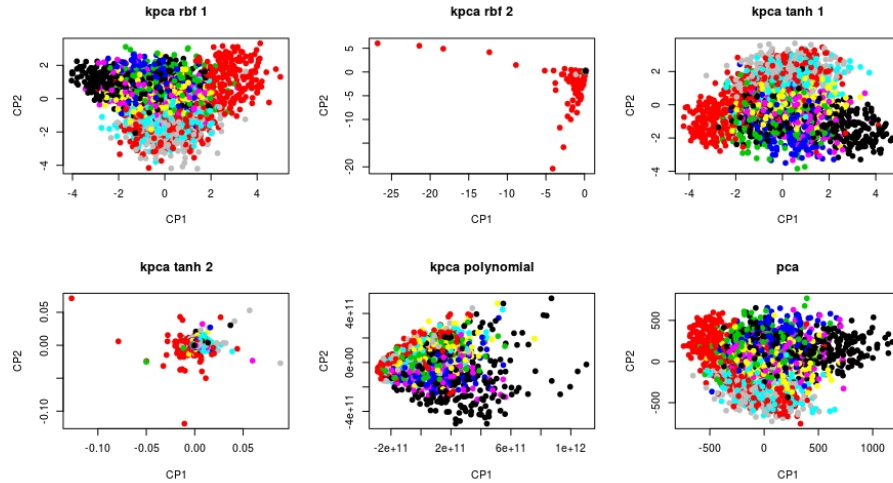


FIGURE 4 – Projection des deux premières CPs selon différents noyaux.

On remarque que certains noyaux sont moins performants pour capturer l'information que sans le bruit, mais que certains sont capables de trouver les relations non-linéaires dans les données bruitées pour retrouver l'information originale.

6 Discussion

On a présenté les fonctions noyau et l'analyse par composantes principales avec noyaux. L'idée de la méthode est la suivante : au lieu de faire une décomposition par valeurs et vecteurs propres de la matrice variance covariance des données \mathbf{x} , on doit faire une décomposition par valeurs et vecteurs propres de la matrice des noyaux. Or, on effectue un ACP sur une projection des données vers un espace où il est possible d'expliquer la variation par les données de manière non-linéaire. On a présenté le truc du noyau : il n'est pas nécessaire de calculer directement les données projetées pour effectuer une ACP car on peut substituer le produit scalaire entre les données par une fonction noyau.

Ces méthodes ont notamment été utilisés dans la détection d'attributs dans la reconnaissance faciale, voir [Kim et al., 2002].

6.1 Avantages

Comme mentionné précédemment, l'ACP avec noyau permet de faire une ACP sur une transformation non-linéaire des données. Les méthodes à noyaux ont

eus beaucoup de succès dans les tâches de classification et de régression car il est possible de trouver un espace où un hyperplan séparateur pourrait parfaitement séparer les données. Dans le contexte de l'ACP, cette méthode peut permettre de découvrir des relations non-linéaires et bien représenter la variabilité des données dans un nombre plus restreint de composantes principales et pourrait aider à mieux interpréter les composantes.

Par exemple, dans le jeu de données jouet, il n'était pas possible de réduire la dimension des données car les relations étaient circulaires, on devait garder toute la dimension des données pour comprendre la distribution des classes. Avec un noyau Gaussien, il était possible de représenter et interpréter les données avec une seule composante.

6.2 Désavantages

Les fonctions à noyau sont très flexibles, mais on doit souvent ajuster des hyperparamètres pour avoir une représentation des données qui est utile pour l'interprétation. Dans notre application, il était simple de distinguer les bonnes projections des mauvaises, car les étiquettes de classe étaient disponibles. Dans le contexte de découverte d'information.

Une propriété intéressante de l'ACP est qu'on peut reconstruire les données en gardant toutes les composantes. Dans notre cas, on ne peut pas les reconstruire car ces données se trouvent dans un espace \mathcal{F} qui n'est pas toujours connu. Avec l'ACP appliqué en vision, il est possible de visualiser l'information conservée. Par exemple, on recrée les images originales avec l'ACP mais en ne conservant que dix composantes principales. On obtient

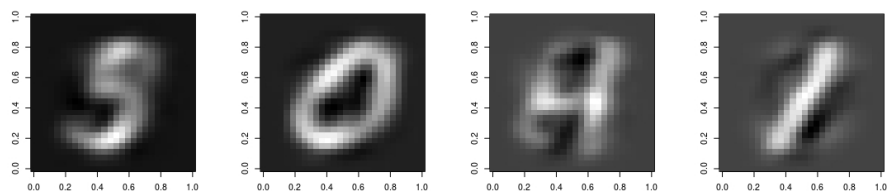


FIGURE 5 – Données reconstruites avec 10 composantes principales

On remarque qu'un humain peut distinguer les chiffres dans la figure 5. Par contre, on ne peut pas faire cette même projection avec les données dans l'espace \mathcal{F} car elles ne sont jamais calculées. Ainsi, la meilleure manière d'évaluer la qualité de l'ACP avec noyau est d'étudier la proportion de l'information des noyaux qui est expliquée par les premières composantes principales.

7 Conclusion

Références

- [Kim et al., 2002] Kim, K. I., Jung, K., and Kim, H. J. (2002). Face recognition using kernel principal component analysis. *IEEE signal processing letters*, 9(2) :40–42.
- [LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324.
- [Muller et al., 2001] Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2) :181–201.
- [Schölkopf et al., 1997] Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer.
- [Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5) :1299–1319.