Algorithme d'apprentissage séquentiel pour la méthode KFD Relations avec la méthode KPCA

Cédric RICHARD, Fahed ABDALLAH

Laboratoire de Modélisation et Sûreté des Systèmes, Université de Technologie de Troyes
12 rue Marie Curie, BP 2060
10010 Troyes cedex, France
cedric.richard@utt.fr, fahed.abdallah@utt.fr

Résumé –
Durant la décennie précédente, de multiples méthodes pour l'analyse et la classification de données fondées sur la théorie des espaces de Hilbert à noyau reproduisant ont été développées. Elles reposent sur le principe fondamental du *kernel trick*, initialement mis en œuvre par Vapnik dans le cadre des Support Vector Machines. Celui-ci permet d'étendre au cas non-linéaire des traitements initialement linéaires en utilisant la notion de noyau. La méthode *Kernel Fisher Discriminant*, nommée KFD, constitue ainsi une généralisation non-linéaire de l'analyse discriminante de Fisher. Bien que son efficacité soit indiscutable, on déplore le fait que sa mise en œuvre nécessite le stockage et la manipulation de matrices de dimension égale au nombre de données traitées, point critique lorsque l'ensemble d'apprentissage est de grande taille. Cet article présente un algorithme séquentiel palliant cette difficulté puisqu'il ne nécessite, ni la manipulation, ni même le stockage de telles matrices. Dans un second temps, un parallèle est proposé entre KFD et KPCA, acronyme de *Kernel Principal Component Analysis*, cette dernière méthode constituant une extension au cas non-linéaire de l'analyse en composantes principales. Les points communs exhibés permettent de proposer un algorithme itératif pour celle-ci également. Il présente de fait des similitudes marquées avec celui dédié à KFD.

Abstract –
In recent years, many detection methods based on the reproducing kernel Hilbert spaces have been developed. The majority of these methods incorporate the fundamental idea used by Vapnik in the Support Vector Machines which consists of extending a linear algorithm for the non-linear case by using the notion of kernels. Kernel Fisher Discriminant (KFD) is one of these methods which leads to competitiveness results in many practical cases. However, use of the KFD method requires storage and handling of matrices of a size equal to that of the number of available examples. This may be critical when the training set is large. This paper presents a sequential algorithm for the KFD method which does not require handling or even the storage of large matrices. In addition, another sequential algorithm, which fulfils the same requirements as that of the KFD algorithm, is presented for the Kernel Principal Component Analysis (KPCA) method which is used to extract non-linear features.

1 Introduction

Le domaine de la Reconnaissance des Formes connaît une révolution depuis le milieu des années 90 avec l'avènement des noyaux reproduisants pour la résolution de problèmes de détection/classification et régression [10, 13]. Ceux-ci permettent en effet de développer un caractère non-linéaire dans nombre de traitements linéaires, sans qu'il soit nécessaire de recourir à d'importants développements théoriques. Aussi l'Analyse en Composantes Principales (ACP) et l'Analyse Factorielle Discriminante (AFD), outils standards en analyse de données, ont-elles été rapidement reformulées afin d'intégrer des caractéristiques non-linéaires. Aujourd'hui, elles sont communément désignées par KPCA [12] et KFD [5], acronymes respectifs de *Kernel Principal Component Analysis* et *Kernel Fisher Discriminant*. Si l'efficacité de ces deux méthodes est indiscutable, elles s'avèrent toutefois délicates à mettre en œuvre lorsque la taille de l'ensemble d'apprentissage est importante. Appliquées à une base regroupant n individus, toutes deux requièrent en effet le stockage et la manipulation de matrices de taille $(n \times n)$. Un algorithme de type EM a été récemment développé afin de remédier à cette situation dans le cadre de la méthode KPCA [4, 7], évolution directe d'une technique séquentielle initialement proposée pour l'ACP [11]. Plus complexe parce qu'elle ne repose pas directement sur la diagonalisation d'une matrice de covariance, la méthode KFD n'a pas encore suscitée autant d'intérêt du point de vue de sa mise en œuvre, exception faite à l'algorithme d'optimisation avec contrainte exposé dans [6].

L'objectif premier de cet article est de remédier à ce manque en proposant un algorithme séquentiel qui ne nécessite pas la manipulation de matrices de grande taille. Celui-ci est ensuite comparé à une méthode séquentielle dédiée à la minimisation de l'erreur quadratique. On rappelle en effet que pour des sorties désirées convenablement choisies, la solution minimisant cette fonction coût est également optimum au sens du critère de Fisher [2, 3]. Un parallèle avec la méthode KPCA est ensuite proposé, montrant au lecteur qu'un algorithme itératif de même type peut également être utilisé dans ce dernier cas. Mais avant, il convient de décrire brièvement le cadre algébrique offert par les espaces de Hilbert à noyau reproduisant, permettant par là-

même de rappeler les propriétés clé ayant contribué au succès des noyaux de Mercer, et d'introduire quelques notations.

2 Espaces à noyau reproduisant et condition de Mercer

Soit $\mathcal{H}$ un espace fonctionnel hilbertien réel de produit scalaire $< \cdot ; \cdot >_{\mathcal{H}}$, composé de fonctions ψ continues sur un ensemble $\mathcal{X}$. D'après le théorème de représentation de Riesz, il existe une fonction unique $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ de la variable $\boldsymbol{x}_i$, étant donné $\boldsymbol{x}_j$ fixé, telle que

$$\psi(\boldsymbol{x}_j) = < \psi \, ; \kappa(\cdot, \boldsymbol{x}_j) >_{\mathcal{H}}, \qquad \forall \psi \in \mathcal{H}. \tag{1}$$

Dans cette expression, $\kappa(\cdot, \boldsymbol{x}_j)$ désigne une fonction définie sur $\mathcal{X}$, obtenue en fixant le second argument de κ à $\boldsymbol{x}_j$. Il en résulte que l'ensemble $\{\kappa(\cdot, \boldsymbol{x}) \ : \ \boldsymbol{x} \in \mathcal{X}\}$ engendre $\mathcal{H}$, et que le produit scalaire $< \cdot ; \cdot >_{\mathcal{H}}$ ne nécessite d'être défini que sur cet ensemble de générateurs. Au vu de cette propriété, κ est appelé *noyau reproduisant* de $\mathcal{H}$. En notant $\phi(\boldsymbol{x})$ la fonction $\kappa(\cdot, \boldsymbol{x})$, l'équation (1) implique

$$< \phi(\boldsymbol{x}_i) \, ; \phi(\boldsymbol{x}_j) >_{\mathcal{H}} = \kappa(\boldsymbol{x}_j, \boldsymbol{x}_i), \tag{2}$$

pour tout $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$. Ce résultat signifie que $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ fournit le produit scalaire des images dans $\mathcal{H}$ de toute paire d'éléments de l'ensemble $\mathcal{X}$. En autorisant l'exploitation de ce concept sans nécessairement connaître explicitement $\mathcal{H}$ et ϕ, la condition de Mercer a contribué aux plus récents développements des structures à noyau [13]. On présente ci-dessous trois noyaux usuels qui vérifient cette condition, ce qui signifie qu'ils fournissent à moindre coup de calcul le produit scalaire des images de deux observations $\boldsymbol{x}_i$ et $\boldsymbol{x}_j$ par une application ϕ. Afin d'élaborer une règle de décision basée sur une statistique polynômiale de degré q, on peut utiliser le noyau reproduisant suivant

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (1 + \boldsymbol{x}_i \cdot \boldsymbol{x}_j)^q. \tag{3}$$

On peut en effet montrer que les composantes de l'application $\phi(\boldsymbol{x})$ associée sont les monômes de degré inférieur ou égal à q constitués des composantes de $\boldsymbol{x}$. Le noyau gaussien

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / \beta_0\right) \tag{4}$$

où β_0 est appelé *largeur de bande*, est un noyau de type radial qui joue un rôle central dans les méthodes d'estimation et de classification à base de noyaux. Enfin, le noyau exponentiel

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\| / \beta_0\right). \tag{5}$$

fournit une surface de décision linéaire par morceaux dans l'espace des observations. D'autres exemples de noyaux de Mercer peuvent être consultés dans [13].

3 Algorithme séquentiel pour KFD

Supposons qu'on dispose d'un ensemble d'apprentissage constitué de n individus $\boldsymbol{x}_k$, se répartissant en n_1 et n_2 représentants pour les deux classes en compétition $\mathcal{C}_1$ et $\mathcal{C}_2$. La méthode KFD consiste en la recherche d'une fonction ψ de $\mathcal{H}$ de sorte que la statistique $\lambda(\boldsymbol{x}) = < \psi \, ; \phi(\boldsymbol{x}) >_{\mathcal{H}}$ maximise le critère de Fisher [3]. En pratique, cette recherche est limitée au sous-espace $\mathcal{H}_n$ engendré par les n fonctions $\{\phi(\boldsymbol{x}_k)\}_{1 \leq k \leq n}$, où les $\boldsymbol{x}_k$ désignent les éléments de la base d'apprentissage. On a ainsi

$$\psi = \sum_{k=1}^{n} \alpha_k \, \phi(\boldsymbol{x}_k),$$

ce qui mène directement à la statistique $\lambda(\boldsymbol{x}) = \boldsymbol{\alpha}^t \, \tilde{\boldsymbol{\kappa}}(\boldsymbol{x})$ avec $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}) = (\kappa(\boldsymbol{x}, \boldsymbol{x}_1) \ \ldots \ \kappa(\boldsymbol{x}, \boldsymbol{x}_n))^t$ et $\boldsymbol{\alpha}$ le vecteur de composantes α_k. Dans ces conditions, on aboutit à la formulation suivante du critère de Fisher, qu'il s'agit de minimiser :

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^t \boldsymbol{N} \boldsymbol{\alpha}}{(\boldsymbol{\alpha}^t \boldsymbol{\mu})^2}. \tag{6}$$

Dans cette expression, $\boldsymbol{N} = \boldsymbol{K}\boldsymbol{K}^t - \sum_{i=1,2} n_i \, \boldsymbol{\mu}_i \, \boldsymbol{\mu}_i^t$, où $\boldsymbol{\mu}_i$ désigne la moyenne des $\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}_k)$ issus de $\mathcal{C}_i$, et $\boldsymbol{\mu} = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$. On rappelle que $\boldsymbol{K}$ désigne la matrice de Gram de terme général $K_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Avant de poursuivre, il convient de remarquer que (6) désigne plus exactement l'inverse du critère de Fisher. Ce choix s'impose par l'expression du gradient $\nabla J(\boldsymbol{\alpha})$ à laquelle on aboutit :

$$\nabla J(\boldsymbol{\alpha}) = \frac{2}{(\boldsymbol{\alpha}^t \boldsymbol{\mu})^2} \left[\boldsymbol{N}\boldsymbol{\alpha} - \frac{\boldsymbol{\alpha}^t \boldsymbol{N} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^t \boldsymbol{\mu}} \boldsymbol{\mu} \right]. \tag{7}$$

Clairement, la norme de $\boldsymbol{\alpha}$ n'influe aucunement la valeur du critère (6). Aussi $\boldsymbol{\alpha}$ peut-il être normalisé de sorte que l'on ait $\boldsymbol{\alpha}^t \boldsymbol{\mu} = 1$. Dans ces conditions et après quelques fastidieux calculs, l'expression (7) mène finalement au terme de mise à jour suivant de $\boldsymbol{\alpha}$ durant son élaboration itérative

$$\begin{aligned} \Delta\boldsymbol{\alpha} = \sum_{i=1}^{n} y_i \left[\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}_i) - y_i \, \boldsymbol{\mu} \right] \\ + n_1 \left(\boldsymbol{\alpha}^t \boldsymbol{\mu}_1\right) \left[\left(\boldsymbol{\alpha}^t \boldsymbol{\mu}_1\right) \boldsymbol{\mu} - \boldsymbol{\mu}_1 \right] \\ + n_2 \left(\boldsymbol{\alpha}^t \boldsymbol{\mu}_2\right) \left[\left(\boldsymbol{\alpha}^t \boldsymbol{\mu}_2\right) \boldsymbol{\mu} - \boldsymbol{\mu}_2 \right], \end{aligned} \tag{8}$$

où l'on a posé $y_i = \boldsymbol{\alpha}^t \, \tilde{\boldsymbol{\kappa}}(\boldsymbol{x}_i)$. L'algorithme séquentiel proposé s'exprime finalement ainsi :

1. initialiser aléatoirement $\boldsymbol{\alpha}$
2. calculer $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ et $\boldsymbol{\mu}$
3. normaliser $\boldsymbol{\alpha}$ selon $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} / (\boldsymbol{\alpha}^t \boldsymbol{\mu})$, puis calculer $\Delta\boldsymbol{\alpha}$
4. rafraîchir $\boldsymbol{\alpha}$ selon $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \eta \, \Delta\boldsymbol{\alpha}$, avec $\eta > 0$
5. retourner en 3. jusqu'à convergence de l'algorithme

On note que cette nouvelle approche ne nécessite, ni le stockage et la manipulation d'une matrice de taille $(n \times n)$ [5], ni l'optimisation d'une fonction quadratique avec $(n + 2)$ contraintes d'égalité [6]. Cette méthode séquentielle s'avère ainsi être une alternative intéressante à l'algorithme KFD original lorsque la base d'apprentissage est de taille importante. Il convient de noter que cet algorithme s'inspire d'une technique itérative proposée dans le cadre de l'analyse discriminante linéaire de Fisher [9]. Elle est elle-même une réminiscence d'une méthode d'apprentissage appelée *règle d'Oja* [8].

4 Expérimentations et comparaisons

La méthode proposée a été expérimentée sur un ensemble de données synthétiques distribuées dans le plan selon deux hyperboloïdes, comme l'indique la figure 1. Ces dernières représentent les classes $\mathcal{C}_1$ et $\mathcal{C}_2$ en compétition, et regroupent chacune 200 individus. La figure 1 montre la frontière de décision obtenue à l'aide d'un noyau exponentiel de largeur de

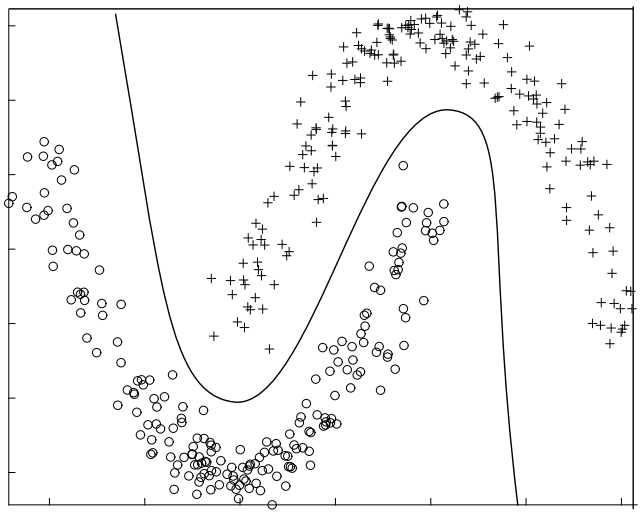

Figure 1: Frontière de décision obtenue grâce à la méthode KFD itérative. Un noyau exponentiel de largeur de bande $\beta_0 = 0.2$ a été utilisé.

bande $\beta_0 = 0.2$. Afin d'apprécier les qualités de convergence de l'algorithme proposé, celui-ci a été comparé à l'algorithme séquentiel proposé pour la méthode KMSE, acronyme de *Kernel Mean Square Error* [2]. Cette dernière a pour vocation d'élaborer des détecteurs à noyau qui minimisent l'erreur quadratique entre sorties obtenues et sorties désirés. Le critère de performance considéré s'exprime dans ce cas ainsi

$$J(\boldsymbol{\alpha}) = \|\boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{y}_d\|^2, \tag{9}$$

où $\boldsymbol{y}_d$ est un vecteur regroupant les sorties désirées, tandis que $\boldsymbol{y} = \boldsymbol{K}\boldsymbol{\alpha}$ représente les sorties obtenues. On rappelle qu'un choix approprié des sorties désirées conduit à une solution optimum au sens du critère de Fisher. En l'occurrence, il s'agit de poser $y_d[i] = n/n_1$ si l'observation $\boldsymbol{x}_i$ est issue de $\mathcal{C}_1$, et $y_d[i] = -n/n_2$ si elle provient de $\mathcal{C}_2$. Le calcul du gradient de $J(\boldsymbol{\alpha})$ mène à

$$\nabla J(\boldsymbol{\alpha}) = \boldsymbol{K}^t\boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{K}^t\boldsymbol{y}_d. \tag{10}$$

Finalement, le terme de mise à jour recherché est donné par

$$\Delta\boldsymbol{\alpha} = \boldsymbol{K}^t\boldsymbol{y} - \boldsymbol{K}^t\boldsymbol{y}_d, \tag{11}$$

que l'on peut écrire pour chaque composante de $\boldsymbol{\alpha}$ ainsi

$$\Delta\alpha_i = \tilde{\boldsymbol{\kappa}}^t(\boldsymbol{x}_i)(\boldsymbol{y} - \boldsymbol{y}_d). \tag{12}$$

L'algorithme séquentiel proposé pour la méthode KMSE prend finalement la forme suivante :

1. initialiser aléatoirement $\boldsymbol{\alpha}$
2. calculer $\boldsymbol{y} = \boldsymbol{K}\boldsymbol{\alpha}$, puis les composantes $\Delta\alpha_i$
3. rafraîchir α_i selon $\alpha_i \leftarrow \alpha_i - \eta\,\Delta\alpha_i$, avec $\eta > 0$
4. retourner en 2. jusqu'à convergence de l'algorithme

La figure 2 compare les vitesses de convergence des algorithmes séquentiels KFD et KMSE, mettant en évidence les qualités intéressantes que présentent l'approche proposée. Ces expérimentations ont été conduites sur le problème représenté par la figure 1, en fixant le pas d'adaptation η à 0.05. Si les critères optimisés (6) et (9) mènent théoriquement à des solutions équivalentes, ils diffèrent toutefois par leur nature. Ceci se traduit par des surfaces d'erreur de types différents, comme cela est montré par [9] dans le cas linéaire. Il en résulte au final des vitesses de convergence différentes.

5 Connections avec la méthode KPCA

La méthode KFD a pour but de concentrer le caractère discriminant des données tandis que la méthode KPCA a trait à leur représentation fidèle. Il est toutefois intéressant de constater que, malgré cette divergence d'objectifs, des techniques comparables peuvent être employées pour résoudre ces deux problèmes. On rappelle que l'extraction d'un axe principal d'inertie par la méthode KPCA consiste en la maximisation du critère suivant [12] :

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^t\boldsymbol{N}\boldsymbol{\alpha}}{(\boldsymbol{\alpha}^t\boldsymbol{K}\boldsymbol{\alpha})}. \tag{13}$$

Le calcul du gradient de $J(\boldsymbol{\alpha})$ mène à l'expression

$$\nabla J(\boldsymbol{\alpha}) = \frac{2}{(\boldsymbol{\alpha}^t\boldsymbol{K}\boldsymbol{\alpha})}\left[\boldsymbol{N}\boldsymbol{\alpha} - \frac{\boldsymbol{\alpha}^t\boldsymbol{N}\boldsymbol{\alpha}}{\boldsymbol{\alpha}^t\boldsymbol{K}\boldsymbol{\alpha}}\,\boldsymbol{K}\boldsymbol{\alpha}\right]. \tag{14}$$

Le vecteur $\boldsymbol{K}\boldsymbol{\alpha}$ a pour composantes $y_i = \boldsymbol{\alpha}^t\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}_i)$. Il est en conséquence noté $\boldsymbol{y}$. Comme dans la section précédente, seule la direction de $\boldsymbol{\alpha}$ importe. Celui-ci peut donc est normalisé de sorte que $\boldsymbol{\alpha}^t\boldsymbol{K}\boldsymbol{\alpha} = 1$, soit $\boldsymbol{\alpha}^t\boldsymbol{y} = 1$. Après quelques calculs destinés à éliminer $\boldsymbol{N}$ dans l'expression (14), on aboutit au terme de mise à jour suivant

$$\Delta\boldsymbol{\alpha} = \sum_{i=1}^{n} y_i\left[\tilde{\boldsymbol{\kappa}}(\boldsymbol{x}_i) - y_i\,\boldsymbol{y}\right] + n\,(\boldsymbol{\alpha}^t\boldsymbol{\mu})\left[(\boldsymbol{\alpha}^t\boldsymbol{\mu})\,\boldsymbol{y} - \boldsymbol{\mu}\right]. \tag{15}$$

On retrouve ainsi l'expression (8) dans laquelle $\boldsymbol{y}$ est venue se substituer à $\boldsymbol{\mu}$, et pour laquelle la distinction entre les deux classes a disparu puisqu'elle n'a plus lieu d'être. Il en résulte des algorithmes semblables, mises à part les différences mineures qui viennent d'être soulignées, ainsi que le choix de la normalisation de $\boldsymbol{\alpha}$.

6 Conclusion et perspectives

Un algorithme séquentiel pour la mise en œuvre de la méthode KFD a été proposé afin de pallier les difficultés pratiques rencontrées lorsque la base d'apprentissage est de grande taille. Une comparaison avec une version séquentielle de la méthode KMSE a permis d'apprécier les qualités de convergence de l'approche proposée. Puis, un parallèle a été effectué avec la méthode KPCA, montrant que des méthodes de résolution équivalentes peuvent être employées. Il apparaît à présent opportun de généraliser l'algorithme séquentiel KFD pour le traitement de problèmes multi-classes [1]. La recherche de plusieurs axes principaux pour la méthode KPCA peut aussi être envisagée.

References

[1] G. BAUDAT, F. ANOUAR. Generalized discriminant analysis using a kernel approach. *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000.

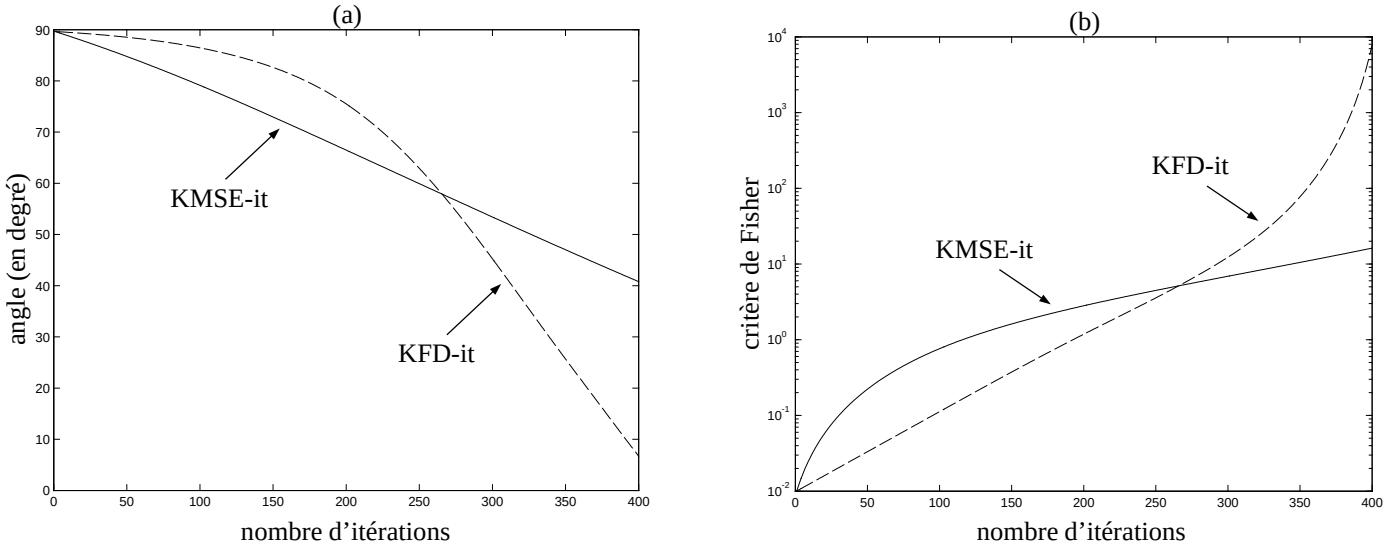

Figure 2: (a) Angle entre les vecteurs α obtenus à l'aide de chacune des deux méthodes itératives considérées et le vecteur optimal au sens du critère de Fisher obtenu par un calcul direct [5], en fonction du nombre d'itérations. (b) Valeur du critère de Fisher pour les solutions obtenues par les méthodes KFD et KMSE itératives, en fonction du nombre d'itérations.

[2] S. A. BILLINGS, K. L. LEE. Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural Networks,* vol. 15, p. 263-270, 2002.

[3] R. O. DUDA, P. E. HART, D. G. STORK. *Pattern Classification*. New York : Wiley and Sons, 2001.

[4] S. MIKA. *Kernalgorithmen zur nichtlinearen Signalverarbeitung in Mermalsrümen*. Master's thesis, Technische Universität Berlin, 1998.

[5] S. MIKA, G. RÄTSCH, J. WESTON, B. SCHÖLKOPF, K. R. MÜLLER. Fisher discriminant analysis with kernels. In Y. H. HU, J. LARSEN, E. WILSON, S. DOUGLAS, (éds). *Proc. Advances in Neural Information Processing Systems*. San Mateo : Morgan Kaufmann, p. 41-48, 1999.

[6] S. MIKA, G. RÄTSCH, K. R. MÜLLER. A mathematical programming approach to the kernel Fisher algorithm. In T. K. LEEN, T. G. DIETTERICH, V. TRESP, (éds). *Proc. Advances in Neural Information Processing Systems*. Cambridge : MIT Press, p. 591-597, 2001.

[7] P. MOERLAND. *Mixture models for unsupervised and supervised learning*. Ph. D. thesis, Ecole Polytechnique Fédérale de Lausanne, 2000.

[8] E. OJA. A simplified neuron model as a principal component analyzer. *Journal of mathematical biology,* vol. 15, p. 267-277.

[9] J. PRINCIPE, D. XU, C. WANG. Generalized Oja's rule for linear discriminant analysis with Fisher criterion. *Proc. ICASSP '97,* vol. 4, p. 3401-3404, Seattle, WA, 1997.

[10] C. RICHARD. *Méthodes à noyau et critères de contraste pour la détection à structure imposée*. Habilitation à diriger des recherches, Université de Technologie de Compiègne, 2002.

[11] S. ROWEIS. EM algorithm for PCA and SPCA. In M. I. JORDAN, M. J. KEARNS, S. A. SOLLA, (éds). *Proc. Advances in Neural Information Processing Systems*. Cambridge, MA : MIT Press, vol. 10, p. 626-632, 1998.

[12] B. SCHÖLKOPF, A. SMOLA, K.-R. MÜLLER. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation,* vol. 10, no. 5, p. 1299-1319, 1998.

[13] V. VAPNIK. *The Nature of Statistical Learning Theory*. New York : Springer-Verlag, 1995.