

Algorithme LMS pour la régression à noyau multivariée parcimonieuse

Pierre-Olivier AMBLARD¹ et Hachem KADRI²

¹GIPSA-lab, CNRS UMR 5216, Université Grenoble Alpes, Grenoble, France

²LIF/CNRS UMR 7279, Aix-Marseille Université, Marseille, France

bidou.amblard@gipsa-lab.inpg.fr, hachem.kadril@lif.univ-mrs.fr

Résumé – Cet article s’intéresse à la régression non-linéaire multivariée par des méthodes à noyau. L’algorithme LMS à noyau (kernel least mean square) est étendu au cas de sorties vectorielles. L’extension est fondée sur la généralisation de la théorie des espaces de Hilbert à noyau reproduisant. Dans ce cadre, nous proposons un critère de cohérence permettant de contrôler la parcimonie de l’algorithme et nous illustrons les performances de l’algorithme et les comparons à celles obtenues par l’algorithme LMS à noyau standard. Nos résultats montrent qu’il est possible d’améliorer les performances de régression des signaux à valeurs vectorielles ou fonctionnelles en tenant compte des dépendances pouvant exister entre les différentes composantes du signal observé, tout en gardant un temps de calcul raisonnable.

Abstract – This paper deals with nonlinear multivariate regression using kernel methods. Specifically, it studies the LMS (least mean square) algorithm in the context where the outputs are vectors and not scalars as usual. In this setting, we develop an extension of the LMS algorithm based on a generalized reproducing kernel Hilbert space theory. We then propose a coherence criterion that controls the sparsity of the algorithm, and we evaluate its performance with respect to the standard kernel LMS algorithm. Our results show that one can improve regression performance for vector-valued or function-valued signals by taking into account the dependencies that may exist between the samples of the signal while keeping a reasonable computational efficiency.

1 Introduction

Nous considérons l’apprentissage d’une fonction $f : \mathcal{X} \rightarrow \mathcal{Y}$ dans un espace de Hilbert à noyau reproduisant (EHNR) dans le cas où l’espace de sortie \mathcal{Y} est un espace de Hilbert de dimension $d > 1$. Ce problème a reçu relativement peu d’attention dans les communautés d’apprentissage automatique et du traitement du signal comparé à son analogue dans le cas de sorties monodimensionnelles ($d = 1$). Notre intérêt pour ce problème vient du fait que dans plusieurs contextes applicatifs les signaux observés sont des signaux à valeurs vectorielles ou fonctionnelles [1, 2]. On peut citer par exemple des images vidéo où à chaque instant l’observation du signal est un vecteur regroupant les pixels de l’image. Il est important de préciser que les méthodes d’apprentissage ou de traitement du signal développées pour des signaux à valeurs réelles (sorties scalaires) peuvent être appliquées à des signaux à valeurs vectorielles mais ont l’inconvénient de traiter indépendamment les composantes du vecteur formé par les échantillons du signal à un instant donné ; ces méthodes ne sont donc pas capables de prendre en compte les dépendances pouvant exister entre les différentes composantes du signal observé.

Motivés par le succès des méthodes à noyau pour le traitement des signaux à valeurs scalaires, nous nous intéressons à la régression multivariée pour des signaux à valeurs vectorielles par des noyaux reproduisants. Le noyau dans ce cadre est une fonction à valeurs opérateurs¹ contrairement aux noyaux standards qui sont des fonctions à valeurs réelles. L’opérateur permet d’incorporer des connaissances a priori sur les sorties vectorielles, et de considérer les corrélations entre

ses composantes. Plus de détails sur les noyaux à valeurs opérateurs et leur utilisation pour l’apprentissage automatique à sorties multiples se trouvent dans les références [3, 4].

Un inconvénient majeur de l’utilisation des noyaux à valeurs opérateurs est leur coût calculatoire élevé. Pour illustrer ceci, considérons le cas où la dimension de l’espace de sortie d est finie et où nous avons à disposition N données d’apprentissage. La sortie de la fonction noyau dans ce cas est une matrice de taille $d \times d$ et la matrice noyau sera donc de taille $Nd \times Nd$ et non pas de taille $N \times N$ comme pour les noyaux à valeurs scalaires. Lorsque la taille du vecteur de sortie est élevée, manipuler et inverser des matrices de taille $Nd \times Nd$ devient problématique. Pour remédier à ce problème, nous nous focaliserons sur l’algorithme LMS à noyau [5]. Cet algorithme a la capacité de traiter les données en ligne et offre donc une complexité de calcul réduite.

Dans la partie 2, nous rappelons les notions de base de la théorie des EHNR avec noyaux à valeurs opérateurs. Dans la partie 3, nous détaillons l’algorithme LMS à noyau à valeurs opérateurs. Nous montrons comment à partir du formalisme général on peut retrouver l’algorithme LMS original, ainsi que d’autres versions de l’algorithme proposées dans la littérature. Nous présentons aussi une généralisation du critère de cohérence pour le cas de noyau à valeurs opérateurs permettant de contrôler la parcimonie de l’algorithme. L’intérêt de la démarche présentée est illustré par des résultats de simulation dans la partie 4. Finalement, des conclusions et perspectives sont énoncées dans la partie 5.

2 Etat de l’art et positionnement

L’algorithme proposé dans cette communication est fondé sur les noyaux reproduisants à valeurs opérateurs et sur la méthode LMS.

1. La fonction noyau est à valeurs matricielles si la dimension d de l’espace de sortie \mathcal{Y} est finie.

Nous rappelons ici brièvement ces deux concepts.

EHNR à valeurs vectorielles. Soient \mathcal{X} et \mathcal{Y} deux espaces de Hilbert et \mathcal{F} un espace d'opérateurs de \mathcal{X} dans \mathcal{Y} , muni d'un produit scalaire $\langle \cdot, \cdot \rangle_{\mathcal{F}}$. $\mathcal{L}(\mathcal{Y})$ (resp. $\mathcal{L}(\mathcal{Y}, \mathcal{X})$) est l'ensemble des opérateurs linéaires bornés de \mathcal{Y} dans \mathcal{Y} (resp. de \mathcal{Y} dans \mathcal{X}).

Définition 1. (EHNR à valeurs vectorielles)

Un espace de Hilbert \mathcal{F} de fonctions de \mathcal{X} dans \mathcal{Y} est un espace de Hilbert à noyau reproduisant s'il existe $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{L}(\mathcal{Y})$ tel que :

- i. la fonction $z \mapsto K(w, z)g$ est dans \mathcal{F} , $\forall z, w \in \mathcal{X}$, $g \in \mathcal{Y}$,
- ii. $\forall f \in \mathcal{F}$, $\langle f, K(w, \cdot)g \rangle_{\mathcal{F}} = \langle f(w), g \rangle_{\mathcal{Y}}$ (propriété reproductrice).

Définition 2. (noyau à valeurs opérateurs)

Un noyau K à valeurs opérateurs est une fonction de $\mathcal{X} \times \mathcal{X}$ dans $\mathcal{L}(\mathcal{Y})$

- i. K est Hermitien si $K(w, z) = K(z, w)^*$, où $K(z, w)^*$ est l'opérateur adjoint de $K(z, w)$,
- ii. K est positif s'il est Hermitien et $\forall r \in \mathbb{N}^+$ et $\forall \{(w_i, u_i)_{i=1}^r\} \in \mathcal{X} \times \mathcal{Y}$, $\sum_{i,j} \langle K(w_i, w_j)u_i, u_j \rangle_{\mathcal{Y}} \geq 0$.

Théorème 1. (bijection entre les EHNR à valeurs vectorielles et les noyaux à valeurs opérateurs)

Un noyau $K(\cdot, \cdot)$ de \mathcal{X} dans $\mathcal{L}(\mathcal{Y})$ est le noyau reproduisant d'un espace de Hilbert \mathcal{F} , si et seulement si K est positif.

Pour la démonstration de ce théorème ainsi que la construction d'un EHNR autour d'un noyau à valeurs opérateurs et l'unicité de cet espace et du noyau, nous renvoyons à [3, 6].

Algorithme LMS linéaire. L'objectif de cet algorithme est d'apprendre une fonction linéaire f de la forme $f : \mathcal{X} \rightarrow \mathbb{R} : f(x) = w^\top x$, $\mathcal{X} \subset \mathbb{R}^p$, à partir d'une séquence d'exemple $\{(x(t), y(t))_{t=1}^N\}$ et de telle sorte à minimiser l'erreur quadratique moyenne $E[(y(t) - w^\top x(t))^2]$. L'algorithme LMS découle de la résolution de ce problème de minimisation au moyen d'un gradient stochastique. A chaque instant $t = 1, \dots, N$, le gradient de l'erreur quadratique moyenne égale à $-2E[e(t)x(t)]$ est estimé à partir des données observées, à savoir $\hat{E}[e(t)x(t)] = e(t)x(t)$ avec $e(t) = y(t) - w(t-1)^\top x(t)$. La mise à jour du vecteur de paramètres w s'écrit alors $w(t) = w(t-1) + \eta e(t)x(t)$, où η est un coefficient appelé pas d'adaptation. En initialisant $w(0) = 0$, l'application répétitive de la mise à jour donne $w(t) = \eta \sum_{i=1}^t e(i)x(i)$ et l'estimation de la sortie à l'instant t est obtenue par

$$\hat{y}(t) = w(t-1)^\top x(t) = \eta \sum_{i=1}^{t-1} e(i)x(i)^\top x(t), \quad t = 1, \dots, N.$$

Travaux connexes et originalité. L'algorithme LMS est un algorithme en ligne puisqu'il traite les données de façon séquentielle. Plusieurs versions de l'algorithme LMS ont été proposées dans la littérature. La version non-linéaire avec des noyaux à valeurs scalaires a été proposé en premier par [5]. D'autres versions de LMS avec noyau existent aussi ; voir par exemple [7]. Il est important de signaler que deux travaux récents se sont intéressés à l'extension des méthodes à noyau en ligne pour des données de sortie multivariées. Dans [8], une méthode multi-noyau LMS a été proposée. Elle est fondée sur l'application de l'algorithme LMS avec plusieurs noyaux où pour chaque composante du vecteur de sortie, un noyau différent est utilisé. Bien

que cette méthode, contrairement à [5], a l'avantage de pouvoir donner plus de poids sur une ou plusieurs composantes du vecteur $y(t)$ par rapport aux autres composantes, elle n'est pas en mesure de considérer des relations entre deux composantes différentes. Ceci est remédié par l'utilisation de noyaux à valeurs opérateurs dans [9] où les auteurs ont proposé une généralisation de l'algorithme NORMA [10] pour l'apprentissage à sorties multiples. Cependant, les auteurs ne se sont pas focalisés sur les aspects de parcimonie de l'algorithme et ont utilisé une procédure naive de troncature afin de réduire la complexité en calcul et en mémoire. Dans ce papier, nous proposons un algorithme LMS non-linéaire et parcimonieux, utilisant les noyaux à valeurs opérateurs et le critère de cohérence, qui a l'avantage de prendre en compte les dépendances pouvant exister entre les sorties multivariées, tout en gardant un temps de calcul raisonnable.

3 Algorithme LMS à noyau multivarié

Nous étendons dans cette partie l'algorithme LMS à noyau aux données de sortie multivariées.

Principe. On s'intéresse à la recherche d'une fonction f dans un EHNR à valeurs vectorielles \mathcal{F} (construit autour d'un noyau à valeurs opérateurs K) minimisant l'erreur quadratique :

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{t=1}^N \|y(t) - f(x(t))\|_{\mathcal{Y}}^2. \quad (1)$$

Une méthode de descente de gradient stochastique est utilisée pour résoudre (4). La mise à jour de la fonction f se fait de façon itérative suivant la formule suivante :

$$f^{(t)} = f^{(t-1)} - \eta \nabla_f \|y(t) - f(x(t))\|_{\mathcal{Y}}^2. \quad (2)$$

En utilisant successivement la linéarité de l'opérateur évaluation, le théorème de représentation de Riesz et la propriété reproductrice des EHNR à valeurs vectorielles, nous obtenons :

$$f^{(t)} = f^{(t-1)} + 2\eta K(x(t), \cdot) [y(t) - f^{(t-1)}(x(t))]. \quad (3)$$

En choisissant $f^{(0)} = 0$, la récursion précédente implique l'existence d'une suite d'éléments $(\alpha_i^{(t)})$ de \mathcal{Y} telle que : $\forall t \geq 0$, $f^{(t)} = \sum_{i=1}^t K(x(i), \cdot) \alpha_i^{(t)}$. On retrouve ainsi la forme de la solution de (4) donnée par le théorème du représentant dans le cas de EHNR à valeurs vectorielles [3, 6]. Cette forme montre que la fonction $f^{(t)}$ a une représentation de dimension finie. La récursion précédente définissant l'algorithme LMS peut donc être remplacée par une récursion définissant la suite des $\alpha_i^{(t)}$, revenant ainsi à travailler directement dans l'espace engendré par la famille $\{K(x(i), \cdot)\}$, $i = 1, \dots, N$. Dans cette optique, le critère à minimiser s'écrit

$$\alpha^N = \arg \min_{\alpha \in \mathcal{Y}^N} \sum_{t=1}^N \|y(t) - \sum_i K(x(t), x(i)) \alpha_i\|_{\mathcal{Y}}^2 \quad (4)$$

où α est le vecteur contenant les α_i . L'algorithme LMS correspondant devient

$$\alpha^t = \begin{pmatrix} \alpha^{t-1} \\ \mathbf{0} \end{pmatrix} + \eta \begin{pmatrix} \mathbf{K}_{t|t-1} \\ K(x(t), x(t)) \end{pmatrix} (y(t) - \mathbf{K}_{t|t-1}^\top \alpha^{t-1}) \quad (5)$$

où $\mathbf{K}_{t|t-1} = (K(x(t), x(1)), \dots, K(x(t), x(t-1)))^\top$ est une matrice d'opérateurs, c'est-à-dire un opérateur de $\mathcal{L}(\mathcal{Y}, \mathcal{Y}^{t-1})$. Le vecteur de paramètre $\alpha^t \in \mathcal{Y}^t$ voit sa taille croître linéairement avec les itérations t . Cette version de l'algorithme est donc pratiquement inutilisable.

Critère de cohérence. L'utilisation des noyaux à valeurs opérateurs donne la possibilité de considérer et d'incorporer les relations entre les sorties dans la phase d'apprentissage. En contrepartie, les matrices de Gram associés à ces noyaux sont de taille élevée. A l'itération t la matrice noyau est de dimension $td \times td$. Réduire la taille de cette matrice tout en gardant l'information essentielle pour l'apprentissage devient primordial. Pour remédier à ce problème, nous proposons une extension aux noyaux à valeurs opérateurs du concept de parcimonie avec critère de cohérence [7]. Ce critère a été proposé pour les noyaux à valeurs scalaires et a l'avantage de conserver les performances de la méthode à noyau tout en présentant un faible coût calculatoire. Le critère de cohérence pour des noyaux normalisés à valeurs scalaires est défini par $\max_{i \in \mathcal{D}_{t-1}} |k(x(i), x(t))| \leq \mu_0$, où \mathcal{D}_{t-1} est un sous-ensemble de $\{1, \dots, N\}$, $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ est le noyau à valeur scalaire et $\mu_0 \in [0, 1]$ est un seuil de cohérence fixé. \mathcal{D}_{t-1} peut être vu comme étant le dictionnaire des noyaux qui a été formé jusqu'à l'itération $t - 1$. La fonction $k(x(t), \cdot)$ est ajoutée au dictionnaire si le critère de cohérence ci-dessus est vérifié, sinon le dictionnaire reste inchangé.

Afin de généraliser le critère aux noyaux à valeurs opérateurs, nous avons besoin de pouvoir comparer les opérateurs $K(x(i), \cdot)$ et $K(x(j), \cdot)$, $\forall i, j = 1, \dots, N$, et former un dictionnaire de noyaux à valeurs opérateurs. Pour cela, nous considérons que le noyau K est un opérateur Hilbert-Schmidt. Il est à noter que cette hypothèse n'est pas très restrictive puisqu'en dimension finie tous les opérateurs sont de Hilbert-Schmidt et en dimension infinie plusieurs opérateurs vérifient cette hypothèse (exemple des opérateurs intégraux associés à des noyaux de carrés intégrables). Si K est un opérateur de Hilbert-Schmidt alors il est de trace finie et pour toute base orthonormale $\{\varphi_l, l \geq 0\}$ de \mathcal{Y} , nous avons

$$\begin{aligned} \langle K(x(i), \cdot), K(x(j), \cdot) \rangle_{HS} &= \sum_l \langle K(x(i), \cdot) \varphi_l, K(x(j), \cdot) \varphi_l \rangle_{\mathcal{F}} \\ &= \sum_l \langle K(x(i), x(j)) \varphi_l, \varphi_l \rangle_{\mathcal{Y}} = \text{Tr}[K(x(i), x(j))], \end{aligned}$$

où Tr désigne l'opérateur trace. Ainsi, le critère de cohérence pour les noyaux à valeurs opérateurs proposé s'écrit

$$\max_{i \in \mathcal{D}_{t-1}} |\text{Tr}[K(x(i), x(t))]| \leq \mu_0. \quad (6)$$

Notons qu'implicitement, l'opérateur est supposé être de norme de Hilbert-Schmidt unité. Dans le cas contraire, il faut normaliser le noyau $K(x(i), \cdot)$ par $\sqrt{\text{Tr}[K(x(i), x(i))]}$.

Algorithme. L'algorithme parcimonieux LMS à noyau à valeur opérateur utilisant le critère de cohérence de parcimonie est décrit par l'algorithme 1. Les différences par rapport à (5) réside essentiellement dans l'ajout ou non d'un élément dans le dictionnaire, et donc dans la définition de la matrice $\mathbf{K}_{t|t-1}$.

Noyaux à valeurs opérateurs séparables et cas particuliers. Nous discutons ici la construction d'un noyau à valeurs opérateurs, et montrons les choix particuliers permettant de retrouver des versions précédentes de LMS proposées dans la littérature. Le choix du noyau à valeurs opérateurs est un problème ouvert qui suscite un intérêt croissant. Les noyaux qui ont été proposés dans la littérature appartiennent majoritairement à la classe de noyaux à valeurs opérateurs séparables et s'écrivent sous la forme :

$$K(\cdot, \cdot) = k(\cdot, \cdot)T, \quad (7)$$

Algorithme 1 LMS à noyau à valeurs opérateurs

Entrée noyau K normalisé, pas $\eta \in \mathbb{R}_+^*$, seuil $\mu_0 \in [0, 1]$

Init. $\alpha^1 = [\eta y(1)]$, $\mathcal{D}_1 = \{1\}$,

A l'instant $t \geq 2$: nouvelles données $(x(t), y(t))$

$$\mathbf{K}_{t|t-1} = (K(x(t), x(i_1)), \dots, K(x(t), x(i_{|\mathcal{D}_{t-1}|})))^\top;$$

$$e(t) = y(t) - \mathbf{K}_{t|t-1}^\top \alpha^{t-1};$$

Si $\max_{i \in \mathcal{D}_{t-1}} |\text{Tr}[K(x(i), x(t))]| > \mu_0$ **alors**

$$\mathcal{D}_t = \mathcal{D}_{t-1};$$

$$\alpha^t = \alpha^{t-1} + \eta \mathbf{K}_{t|t-1} e(t);$$

Sinon $\mathcal{D}_t = t \cup \mathcal{D}_{t-1}$

$$\alpha^t = \begin{pmatrix} \alpha^{t-1} \\ \mathbf{0} \end{pmatrix} + \eta \begin{pmatrix} \mathbf{K}_{t|t-1} \\ K(x(t), x(t)) \end{pmatrix} e(t);$$

Fin Si

$$f^{(t)}(x(t)) = (K(x(t), x(i_1)), \dots, K(x(t), x(i_{|\mathcal{D}_t|})))^\top \alpha^t.$$

où k est un noyau à valeurs scalaires et T est un opérateur dans $\mathcal{L}(\mathcal{Y})$. Un tel noyau se décompose donc en deux : (1) un noyau standard qui dépend des entrées et modélise la relation non-linéaire entre eux et les sorties, et (2) un opérateur qui agit sur les sorties multivariées $y \in \mathcal{Y}$ pour tenir en compte les différentes dépendances entre les composantes des sorties.

Il est important de mentionner que si on construit K suivant (7) utilisant le noyau linéaire ($k(\cdot, \cdot) = \langle \cdot, \cdot \rangle$) et l'opérateur identité ($T = Id$), on retrouve l'algorithme LMS linéaire standard décrit dans la Section 2.2. Avec un noyau non-linéaire k et l'opérateur identité, l'algorithme 1 devient le LMS à noyau de [5]. Aussi, l'algorithme LMS à noyaux multiples proposé dans [8] peut-être vu comme un cas particulier de l'algorithme 1. En effet, si on utilise l'algorithme 1 avec le noyau K tel que $K(\cdot, \cdot) = \text{diag}[k_m(\cdot, \cdot)]_{m=1}^d$,² on retrouve le LMS multi-noyau de [8].

4 Une illustration expérimentale

L'algorithme proposé a été testé sur données simulées pour différents types de noyaux à valeurs opérateurs et nous présentons deux scénarii à titre d'exemple. Nous nous intéressons à la prédiction non-linéaire de séries temporelles multivariées. Nous considérons des modèles du type couplage de Glass-Mackey. La série temporelle bivariée utilisée est définie par :

$$\begin{aligned} z_t &= z_{t-1} - 0.4 \left(z_{t-1} - \frac{2z_{t-4}}{1 + z_{t-4}^{10}} \right) y_{t-5} + 0.3w_{t-3} + \varepsilon_{z,t} \\ w_t &= 0.6w_{t-1} + \frac{0.8w_{t-2}}{1 + w_{t-2}^{10}} + 0.4z_{t-2} + \varepsilon_{w,t}, \end{aligned}$$

où $\varepsilon_{z,t}$ et $\varepsilon_{w,t}$ sont des bruits gaussiens indépendants de moyenne nulle et de variance égale à 10^{-2} . L'objet est dans cet exemple de prédire le signal $y(t) = (z_t, w_t)^\top$ à partir de son passé.

Dans le **scénario 1**, on utilise 5 échantillons temporels passés de la série bivariée pour prédire. Dans les notations des paragraphes précédents, les données d'apprentissage sont donc $\{x(i) = (z_t, w_t)_{t=i-4}^i; y(i) = (z_{i+1}, w_{i+1})_{i \geq 5}\}$. L'algorithme 1 est utilisé sur ces données pour produire une estimation de $y(i)$. Les noyaux à valeurs opérateurs qui sont utilisés sont des noyaux séparables de la forme (7). Plus précisément, nous considérons le noyau à valeur scalaire $k(x, y) = \exp(-\|x - y\|^2)$ associé à : **1**. L'opérateur identité.

2. L'opérateur $\text{diag}[u_m]_{m=1}^d$ permet de construire une matrice diagonale de taille $d \times d$ où les valeurs u_1, \dots, u_m sont stockées sur la diagonale.

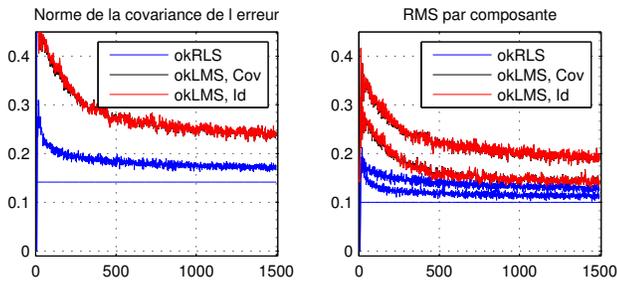


FIGURE 1 – Gauche : Norme de la covariance de l’erreur de prédiction pour le kRLS à valeurs opérateurs [12], le kLMS dans les cas $T =$ covariance de la sortie, $T = I$. Droite : Dans les mêmes cas, écart type de l’erreur pour chaque composante. Les constantes représentent les bornes inférieures. Les courbes rouges et noires sont superposées.

Ceci revient à utiliser des algorithmes LMS à noyau en parallèle [5] ; **2.** L’opérateur diagonal où des poids différents sont utilisés dans la diagonale³. Ceci revient à utiliser l’algorithme LMS à noyaux multiples [8] ; **3.** L’opérateur de covariance. On utilise la covariance empirique calculé à partir de $(y(i))_{i=1}^t$ comme estimateur de la covariance de y ; **4.** L’opérateur multi-tâche [11]. Des poids élevés sont données sur la diagonale pour la relation d’une composante avec elle même, et des poids moins élevés pour la relation entre deux composantes différentes⁴.

La figure 1 montre la performance du kLMS à valeurs opérateurs dans les cas où T est l’opérateur covariance ou l’identité. Nous comparons également aux performances obtenues par le kRLS à valeurs opérateurs [12], et aux bornes inférieures du problème. Les paramètres sont choisis de sorte à ce que chaque algorithme utilise asymptotiquement 90 régresseurs en moyenne. Dans l’exemple simple étudié, la corrélation entre les composantes n’est pas suffisamment forte pour que le choix de l’opérateur covariance pour T améliore significativement les résultats obtenus avec $T = Id$. Par contre les deux autres choix d’opérateurs T sont clairement plus mauvais, comme illustré dans le tableau 1 (scenario 1), où l’on montre les performances asymptotiques obtenues (moyenne sur les 500 derniers échantillons temporels et sur 500 réalisations).

Le **scénario 2** est plus difficile. On essaie de prédire la sortie bivariable uniquement à partir du présent de la première composante z . Les données d’apprentissage sont donc $\{x(i) = z_i; y(i) = (z_{i+1}, w_{i+1})\}_{i \geq 1}$. Les paramètres utilisés assurent que tous les algorithmes utilisent asymptotiquement 65 régresseurs en moyenne. Le tableau 1 (scenario 2) montre les performances asymptotiques obtenues (moyenne sur les 500 derniers échantillons temporels et sur 100 réalisations). Sans que l’amélioration soit exceptionnelle, le noyau séparable utilisant la covariance de la sortie à prédire conduit à des résultats meilleurs que le noyau identité, et que les autres types de noyaux.

5 Conclusion

Dans cette communication, nous présentons un algorithme LMS pour la régression à noyau multivariée parcimonieuse. L’algorithme est une généralisation de l’algorithme LMS non-linéaire pour des signaux à valeurs vectorielles ou fonctionnelles. Il utilise des noyaux à valeurs opérateurs qui permettent de considérer les dépendances entre les composantes du vecteur de sortie. Nous avons étendu le critère de

3. Le poids 0.8 est donnée pour la première composante et 0.2 pour la deuxième. Plus de poids est donnée à la première composante puisqu’il y a plus de relation entre x_t et x_{t+1} qu’avec x_t et y_{t+1} .

4. On utilise le poids 0.8 sur la diagonale et 0.2 hors diagonale.

TABLE 1 – Performance (moyenne et déviation standard de l’erreur quadratique moyenne) de l’Algorithme LMS à noyaux à valeurs opérateurs avec les opérateurs : identité, diagonale, covariance et multi-tâche.

Scenario 1, Algorithmes	Ecart type de l’erreur
LMS à noyau (Identité)	0.260 ± 0.014
LMS multi-noyau (Diagonale)	0.368 ± 0.018
LMS à noyau opérateur (Covariance)	0.260 ± 0.013
LMS à noyau opérateur (Multi-tâche)	0.277 ± 0.014
RLS à noyau opérateur (Covariance)	0.179 ± 0.009

Scenario 2, Algorithmes	Ecart type de l’erreur
LMS à noyau (Identité)	0.374 ± 0.014
LMS multi-noyau (Diagonale)	0.429 ± 0.017
LMS à noyau opérateur (Covariance)	0.366 ± 0.013
LMS à noyau opérateur (Multi-tâche)	0.376 ± 0.014
RLS à noyau opérateur (Covariance)	0.284 ± 0.013

cohérence à ce cadre afin que l’algorithme proposé présente un faible coût calculatoire tout en conservant des bonnes performances de prédiction. Des expériences toutes limitées dans un problème simple de prédiction de séries temporelles montrent non seulement l’intérêt de l’approche multivariée, mais également l’importance de prendre en compte les dépendances entre les sorties via les noyaux à valeurs opérateurs. Toutefois, l’optimisation des paramètres doit être incluse, et l’analyse de l’algorithme reste à faire. L’important est de notamment étudier la vitesse de croissance du dictionnaire. Reste-t-il de taille finie pour des entrées bornées comme dans le cas scalaire [7] ?

Références

- [1] J. O. Ramsay and B.W. Silverman. Applied functional data analysis : methods and case studies. *Springer New York*, 2002.
- [2] P. O. Amblard and O. Michel. Causalité de Granger pour des signaux à valeurs fonctionnelles. *XXIVème colloque GRETSI*, 2013.
- [3] C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.
- [4] M. A. Álvarez, L. Rosasco and N. D. Lawrence. Kernels for vector-valued functions : a review. *Foundation and Trends in Machine Learning*, 4(3):195–266, 2012.
- [5] W. Liu, P. P. Pokharel and J. C. Principe. The kernel least-mean-square algorithm. *IEEE Trans. on Signal Processing*, 56(2):543–554, 2008.
- [6] H. Kadri, E. Duflos, P. Preux, S. Canu, and M. Davy Nonlinear functional regression : a functional RKHS approach. *Artificial Intelligence and Statistics (AISTATS)*, 2010.
- [7] C. Richard, J. C. M. Bermudez, and P. Honeine. Online prediction of time series data with kernels. *IEEE Trans. on Signal Processing*, 57(3):1058–1067, 2009.
- [8] F. A. Tobar, S. Y. Kung, and D. P. Mandic Multikernel least mean square algorithm. *IEEE Trans. on Neural Networks*, 25(2):265–277, 2014.
- [9] J. Audiffren and H. Kadri Online learning with operator-valued kernels. *European Symposium on Artificial Neural Networks (ESANN)*, 2015.
- [10] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Trans. on Signal Processing*, 52(8):2165–2176, 2004.
- [11] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 2005.
- [12] P. O. Amblard and H. Kadri Operator-valued Kernel Recursive Least Squares Algorithm *EUSIPCO*, 2015.