OKVAR : une nouvelle famille de modèles autorégressifs vectoriels à noyaux à valeurs matricielles

Néhémy LIM[1,2], Cédric AULIAC[2], Florence D'ALCHÉ-BUC[3]

[1]IBISC EA 4526, Université d'Évry-Val d'Essonne
23 bd de France, 91037 Évry Cedex, France

[2]CEA, LIST
91191 Gif-sur-Yvette Cedex, France

[3]Institut Mines-Télécom, Télécom ParisTech, LTCI CNRS UMR 5141
37-39 rue Dareau, 75014 Paris, France

nlim@ibisc.univ-evry.fr, cedric.auliac@cea.fr, florence.dalche@telecom-paristech.fr

Résumé – En analyse de séries temporelles multivariées, la problématique de la prévision consiste à estimer les valeurs futures du système étudié à partir d'un historique de vecteurs d'états observés dans le passé. Pour attaquer ce problème canonique, nous définissons dans ce travail une nouvelle famille de modèles autorégressifs vectoriels non paramétriques construits à partir de noyaux à valeurs matricielles. Pour les apprendre, nous avons recours à des méthodes proximales qui permettent d'estimer les paramètres du modèle sous contrainte de parcimonie. Afin de démontrer l'efficacité de nos modèles, nous les appliquons à des données simulées, obtenues à partir de systèmes dynamiques non linéaires présentant des structures topologiques variées.

Abstract – In multivariate time series analysis, forecasting consists in estimating future values of the observed system based on previously observed values. In order to address this issue, we define in this work a new family of nonparametric vector autoregressive models based on matrix-valued kernels. To learn such models, we resort to proximal methods appropriate to estimate the model parameters under sparsity constraints. In order to emphasize the performance of the developed models, we apply them on simulated data, obtained using non linear dynamical systems exhibiting various topological structures.

1 Introduction

Les données issues d'un système dynamique résultent en général de mesures de plusieurs caractéristiques, appelées *variables d'état*, effectuées à plusieurs points de temps espacés régulièrement sur un laps de temps fini. Si l'on note d le nombre de variables d'état et $N+1$ le nombre de points de temps de mesure, ces données prennent alors la forme d'une série temporelle multivariée de dimension d et de longueur $N+1$, notée $\mathbf{x}_{1:N+1} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N+1}\} \subseteq \mathbb{R}^d$ où $\mathbf{x}_\ell$ désigne l'état du système à l'instant $t_\ell, \ell \in \mathbb{N}_{N+1}$. Classiquement, on admet que l'évolution de l'état d'un système dynamique est régie par une fonction h, telle que $\mathbf{x}_{t+1} = h(\mathbf{x}_t, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t+1-p}) + \mathbf{u}_{t+1}$ où t est une mesure discrète du temps et $\mathbf{u}_{t+1}$ est un terme de bruit centré. On dit alors que h est un modèle autorégressif vectoriel d'ordre p. Nous faisons ici l'hypothèse que le processus sous-jacent est stationnaire et Markovien d'ordre 1, c'est-à-dire $p = 1$. Dans le cadre de l'apprentissage supervisé, le problème de l'autorégression consiste à inférer une hypothèse $\hat{h} : \mathbb{R}^d \to \mathbb{R}^d$ à partir d'un ensemble d'apprentissage $\mathcal{S}_N$ formé à partir des vecteurs d'état observés : $\mathcal{S}_N = \{(\mathbf{x}_\ell, \mathbf{x}_{\ell+1})\}_{\ell=1}^N \subseteq \mathbb{R}^d \times \mathbb{R}^d$. Le modèle autorégressif vectoriel le plus populaire à ce jour est le modèle VAR(1) : $h(\mathbf{x}_t) = A\mathbf{x}_t$ où A est une matrice de taille $d \times d$. Toutefois, le caractère non linéaire de la dynamique de la plupart des systèmes réels rencontrés limite l'intérêt des modèles linéaires. À défaut de connaissances expertes qui imposeraient une forme *a priori* au modèle, il est d'usage de recourir à des méthodes non paramétriques. Partant de ces hypothèses et observations, nous proposons de définir une nouvelle famille de modèles autorégressifs vectoriels non paramétriques, OKVAR (*Operator-valued Kernel-based Vector AutoRegressive*) [6] fondés sur la théorie des espaces de Hilbert à noyaux autoreproduisants (RKHS), à valeurs opérateurs [8], appropriée pour l'apprentissage de fonctions à valeurs vectorielles [7]. Ces modèles étendent les modèles VAR(1) au cas non linéaire. Ces modèles dépendant directement des données observées, leur parcimonie se mesure en nombre de données réellement utilisées pour les définir. Nous proposons d'incorporer une pénalité de norme mixte pour imposer ce type de parcimonie et nous développons un algorithme proximal pour l'estimation des paramètres de ces modèles sous contraintes de parcimonie adaptée. D'abord présentés dans le cadre d'une application particulière, l'inférence de réseaux à partir de cinétiques d'observations, nous considérons ici le cas général de la modélisation de séries temporelles et nous illustrons la famille

OKVAR et son algorithme d'apprentissage sur un jeu de données.

2 La famille de modèles OKVAR

La théorie classique des RKHS à noyaux à valeurs scalaires offre un cadre rigoureux pour la régression pénalisée de fonctions à valeurs scalaires. Dans le cas de fonctions à valeurs vectorielles, c'est la théorie des RKHS fondée sur des noyaux à valeurs opérateurs, issue des travaux précurseurs du début des années 1970 [8], qui est pertinente. En apprentissage, les premières applications de cette théorie portent sur des problèmes multi-tâches [4, 7]. C'est actuellement un domaine de recherche actif (cf. [1] et références incluses). Cependant, à notre connaissance, l'utilisation des noyaux à valeurs opérateurs dans le contexte des séries temporelles est nouveau. Nous introduisons à présent brièvement quelques éléments de la théorie des RKHS pour les fonctions à valeurs vectorielles.

Définition 1 (Noyau à valeurs opérateurs [8, 7])**.** *Soient un ensemble $\mathcal{X}$ et un espace de Hilbert $\mathcal{Y}$, muni du produit scalaire $\langle\cdot,\cdot\rangle_{\mathcal{Y}}$. Un noyau semi-défini positif à valeurs dans $\mathcal{L}(\mathcal{Y})$ est une fonction $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ telle que :*

— $\forall(\mathbf{x}, \mathbf{z}) \in \mathcal{X} \times \mathcal{X},\ K(\mathbf{x}, \mathbf{z}) = K(\mathbf{z}, \mathbf{x})^*$

— $\forall m \in \mathbb{N}, \forall\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m \subseteq \mathcal{X}\times\mathcal{Y},\ \sum_{i,j=1}^m \langle \mathbf{y}_i, K(\mathbf{x}_i, \mathbf{x}_j)\mathbf{y}_j\rangle_{\mathcal{Y}} \geq 0$

où $\mathcal{L}(\mathcal{Y})$ désigne l'ensemble des opérateurs linéaires bornés de $\mathcal{Y}$ dans lui-même et on note A^* l'opérateur adjoint de $A \in \mathcal{L}(\mathcal{Y})$.

Dans le contexte de l'autorégression vectorielle, $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ et $\mathcal{L}(\mathbb{R}^d) = \mathbb{R}^{d\times d}$ est l'ensemble des matrices de taille $d \times d$. On dit alors que K est un noyau à valeurs matricielles. Dans ce travail, nous étudions les modèles autorégressifs vectoriels non paramétriques définis par :

$$h = \sum_{\ell=1}^{N} K(\cdot, \mathbf{x}_\ell)\mathbf{c}_\ell \tag{1}$$

où $\mathbf{x}_{1:N+1} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N+1}\} \subseteq \mathbb{R}^d$ est la série temporelle des vecteurs d'états observés, $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ est un noyau à valeur matricielle et $\{\mathbf{c}_\ell\}_{\ell=1}^N \subseteq \mathbb{R}^d$ sont des paramètres à estimer. Nous appelons modèle Vectoriel Autorégressif à base de Noyau à valeur Opérateur (en anglais Operator-valued Kernel-based Vector Autoregressive, OKVAR) tout modèle vectoriel autorégressif de la forme donnée par l'équation (1) [6]. Par la suite, nous nous intéressons à une famille particulière de noyaux définie ci-après :

Proposition 1 (Noyau décomposable [7, 3])**.** *Si $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ est un noyau scalaire et $B \in \mathbb{S}_+^d$ est une matrice semi-définie positive de $\mathbb{R}^{d\times d}$, alors la fonction $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ définie par :*

$$\forall(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^d \times \mathbb{R}^d,\ K(\mathbf{x}, \mathbf{z}) = k(\mathbf{x}, \mathbf{z})B \tag{2}$$

est un noyau à valeurs matricielles.

Les noyaux de la forme (2) sont appelés noyaux *décomposables* et ont notamment été utilisés dans le contexte de l'apprentissage multi-tâches [4]. La matrice B encode les dépendances entre les tâches ou sorties. L'exemple le plus simple consiste à choisir $B = Id$ la matrice identité de taille $d \times d$. Dans ce cas, le coefficient $(p, q) \in \mathbb{N}_d^2$ de la matrice $K(\mathbf{x}, \mathbf{z})$ est $K(\mathbf{x}, \mathbf{z})_{pq} = k(\mathbf{x}, \mathbf{z})\delta_{pq}$, on considère alors que toutes les tâches à apprendre sont très différentes les unes des autres et toutes les sorties sont en conséquence traitées de manière indépendante. Comme nous faisons l'hypothèse que les séries temporelles d'intérêt sont non linéaires, nous nous concentrons sur des noyaux non linéaires. Nous notons $k_{\text{Gauss}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ le noyau gaussien scalaire : $k(\mathbf{x}, \mathbf{z}) = \exp(-\gamma\|\mathbf{x} - \mathbf{z}\|_2^2)$ pour $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$ avec $\gamma > 0$. Nous définissons ensuite le noyau décomposable non linéaire suivant.

Définition 2 (Noyau décomposable gaussien)**.** *On appelle* noyau décomposable gaussien *et on note $K_{dec} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ tout noyau matriciel décomposable de la forme*

$$\forall(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^d \times \mathbb{R}^d,\ K_{dec}(\mathbf{x}, \mathbf{z}) = k_{Gauss}(\mathbf{x}, \mathbf{z})B \tag{3}$$

avec $B \in \mathbb{S}_+^d$

Définition 3 (Modèle OKVAR décomposable gaussien)**.** *Soit une série temporelle $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N+1}\} \subseteq \mathbb{R}^d$, le modèle OKVAR décomposable gaussien noté h_{dec} associé au noyau décomposable gaussien est défini par :*

$$h_{dec} = \sum_{\ell=1}^{N} K_{dec}(\cdot, \mathbf{x}_\ell)\mathbf{c}_\ell \tag{4}$$

3 Apprentissage d'un modèle OKVAR

Régression pénalisée. Nous supposerons ici que le noyau à valeurs matricielles K a été spécifié (se référer à [6] pour l'apprentissage d'un noyau). Par exemple, pour un noyau décomposable gaussien, la largeur de bande γ peut avoir été fixée ou sélectionnée et la matrice B, fournie *a priori* comme une donnée du problème. Dans ce cas, apprendre un modèle OKVAR revient à apprendre les paramètres $\{\mathbf{c}_\ell\}_{\ell=1}^N \subseteq \mathbb{R}^d$ du modèle. Nous notons $C \in \mathbb{R}^{d\times N}$, la matrice qui contient les N vecteurs $\mathbf{c}_\ell$ rangés en colonnes et h_C, le modèle correspondant. Pour estimer C, nous cherchons à minimiser la fonction de coût régularisée suivante :

$$\mathcal{J}(C) = \sum_{t=1}^{N} \|\mathbf{x}_{t+1} - h_C(\mathbf{x}_t)\|_2^2 + \Omega(h_C) \tag{5}$$

où $\Omega(h_C)$ est la somme de deux termes : $\Omega(h_C) = \lambda_h \|h_C\|_{\mathcal{H}_K}^2 + \Omega_C(C)$ avec :

— $\|h_C\|_{\mathcal{H}_K}^2 = \sum_{i,j=1}^N \mathbf{c}_i^T K(\mathbf{x}_i, \mathbf{x}_j)\mathbf{c}_j$. Cette norme joue le rôle d'une norme ℓ_2 sur C pondérée par les matrices $K(\mathbf{x}_i, \mathbf{x}_j)$ et correspond au terme classique de régularisation utilisé pour éviter le surapprentissage.

— $\Omega_C : \mathbb{R}^{d\times N} \to \mathbb{R}_+ \cup \{+\infty\}$ est un terme de régularisation supplémentaire que nous analysons par la suite.

Lorsque $\Omega_C(C) = 0$, minimiser (5) revient à résoudre le problème de la Kernel Ridge Regression. Dans ce cas, la matrice de paramètres C peut être calculée analytiquement :

$$\mathbf{c} = (\mathbf{K} + \lambda_h Id)^{-1} \mathbf{x}_{2:N+1} \tag{6}$$

où $\mathbf{c} \in \mathbb{R}^{Nd}$ est la forme vectorisée de la matrice C obtenue en empilant les vecteurs $\mathbf{c}_\ell$, $\mathbf{K} = (K(\mathbf{x}_\ell, \mathbf{x}_t))_{\ell,t=1}^N \in \mathbb{R}^{Nd \times Nd}$ est la matrice de Gram et $\mathbf{x}_{2:N+1} \in \mathbb{R}^{Nd}$ est le vecteur résultant de la concaténation des observations $\{\mathbf{x}_\ell\}_{\ell=2}^{N+1}$. Cependant, la solution donnée par (6) n'est pas parcimonieuse. Afin d'induire de la parcimonie dans le modèle, on peut introduire une pénalité en norme ℓ_1 sur C en posant $\Omega_{\ell_1}(C) = \lambda_C \|C\|_1$ avec $\lambda_C > 0$ et où $\|\cdot\|_1$ désigne à la fois la norme ℓ_1 d'un vecteur et celle de la forme vectorisée d'une matrice. Dans les approches non paramétriques, une problématique-clé est celle du contrôle de la complexité. Une manière de l'aborder consiste à n'autoriser qu'un nombre restreint de paramètres $\mathbf{c}_\ell$ non nuls, ce qui signifie que seule une poignée de données est réellement impliquée dans le modèle. En analogie aux SVMs, nous utilisons le vocable de *vecteurs de support* pour désigner les données correspondant à des vecteurs $\mathbf{c}_\ell$ non nuls. Une régularisation par la norme ℓ_1 pénalise de manière uniforme toutes les coordonnées des vecteurs $\mathbf{c}_\ell$ mais ne permet pas d'annuler tous les coefficients d'un vecteur $\mathbf{c}_\ell$ donné. Pour y parvenir, une stratégie fondée sur une parcimonie *structurée* est plus appropriée en considérant les colonnes de C, c'est-à-dire les vecteurs $\mathbf{c}_\ell$, comme une partition des coefficients de la matrice. Une contrainte de ce type qu'on note Ω_{ℓ_1/ℓ_2} prend la forme suivante :

$$\Omega_{\ell_1/\ell_2}(C) = \lambda_C \sum_{\ell=1}^{N} w_\ell \|\mathbf{c}_\ell\|_2 \tag{7}$$

Ω_{ℓ_1/ℓ_2} est la norme mixte ℓ_1/ℓ_2 [9]. Cette norme possède plusieurs caractéristiques intéressantes : elle agit comme une norme ℓ_1 au niveau des groupes (les vecteurs $\mathbf{c}_\ell$) tandis qu'au sein de chaque vecteur $\mathbf{c}_\ell$ les coefficients sont sujets à une contrainte en norme ℓ_2. Les valeurs des poids $\{w_\ell\}_{\ell=1}^N \subseteq \mathbb{R}_+$ dépendent de l'application.

Algorithme proximal. Dans le cas où Ω_C est Ω_{ℓ_1} ou Ω_{ℓ_1/ℓ_2}, (5) est une fonction de coût convexe qui est la somme de deux termes :

$$\mathcal{J}(C) = f_C(\mathbf{c}) + g_C(\mathbf{c}) \tag{8}$$

où

— $f_C(\mathbf{c}) = \sum_{t=1}^N ||h_C(\mathbf{x}_t) - \mathbf{x}_{t+1}||^2 + \lambda_h \|h_C\|^2_{\mathcal{H}_K}$

— $g_C(\mathbf{c}) = \Omega(C)$

On remarque que f_C et g_C sont deux fonctions convexes de C, en particulier f_C est différentiable alors que g_C est non lisse mais est sous-différentiable. La décomposition (8) de la fonction de coût $\mathcal{J}(C)$ justifie l'emploi d'un algorithme à base de gradients proximaux. L'Algorithme 1 repose sur les éléments suivants :

— L_C est une constante de Lipschitz de $\nabla_C f_C$, le gradient de f_C par rapport à la variable C

— les variables intermédiaires $t^{(m)}$ et $\mathbf{y}^{(m)}$ introduites respectivement aux étapes 2 et 3 permettent d'accélérer la méthode [2]

Algorithme 1 Minimiser (5)

Entrées : $\mathbf{c}^{(0)} \in \mathbb{R}^{Nd}$; M ; $\epsilon_\mathbf{c}$; L_C
Initialisation : $m = 0; \mathbf{y}^{(1)} = \mathbf{c}^{(0)}; t^{(1)} = 1$; `stop` :=`faux`
tant que $m < M$ et `stop=faux` **faire**
  0. : $m := m + 1$
  1. : $\mathbf{c}^{(m)} = \text{prox}_{\frac{1}{L_C} g_C} \left(\mathbf{y}^{(m)} - \frac{1}{L_C} \nabla_{\mathbf{y}^{(m)}} f_C(\mathbf{y}^{(m)}) \right)$
  si $||\mathbf{c}^{(m)} - \mathbf{c}^{(m-1)}|| \leq \epsilon_\mathbf{c}$ **alors**
    `stop` :=`vrai`
  sinon
    2. : $t^{(m+1)} = \frac{1+\sqrt{1+4t^{(m)\,2}}}{2}$
    3. : $\mathbf{y}^{(m)} = \mathbf{c}^{(m)} + \frac{t^{(m)}-1}{t^{(m+1)}} \left(\mathbf{c}^{(m)} - \mathbf{c}^{(m-1)} \right)$
  fin si
fin tant que
Sortie : $\mathbf{c}^{(m)}$

L'étape de gradient proximal à l'itération m est donnée par :

$$\begin{aligned} & \text{prox}_{\frac{1}{L_C} g_C} \left(\mathbf{y}^{(m)} - \frac{1}{L_C} \nabla_{\mathbf{y}^{(m)}} f_C(\mathbf{y}^{(m)}) \right)_{I_\ell} \\ = \; & \mathcal{T}_{s_\ell} \left(\mathbf{y}^{(m)} - \frac{1}{L_C} \nabla_{\mathbf{y}^{(m)}} f_C(\mathbf{y}^{(m)}) \right)_{I_\ell} \end{aligned}$$

où pour $s \geq 0$, $\mathcal{T}_s$ désigne l'opérateur de seuillage doux. Spécifiquement, lorsque $g_C = \Omega_{\ell_1/\ell_2}$, $s_\ell = \frac{\lambda_C w_\ell}{L_C}$ et $I_\ell,\ \ell \in \mathbb{N}_N$ est le sous-ensemble des indices correspondant à la ℓ-ème colonne de la matrice C. Dans le cas où $g_C = \Omega_{\ell_1}$, $s_\ell = \frac{\lambda_C}{L_C}$ et $I_\ell,\ \ell \in \mathbb{N}_{Nd}$ est réduit à un singleton correspondant à un coefficient donné de C.

4 Résultats

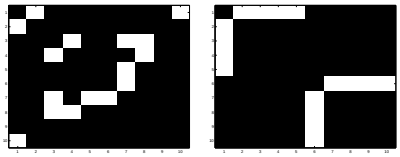
Génération des données. Nous présentons ici les performances de la famille de modèles OKVAR (Table 1) et du modèle VAR(1) à travers plusieurs expériences numériques. Les systèmes dynamiques ont été construits à partir d'une structure de graphe orienté indiquant quelles variables influencent quelles autres. Pour cela, nous avons considéré deux motifs distincts de structure de graphe : « random » et « hub » (Figure 1). Ces graphes ont été obtenus via le package `flare` de `R`. Une série multivariée non linéaire de dimension d est alors générée selon le modèle :

$$\begin{cases} \mathbf{x}_1 \sim \mathcal{N}(0, \Sigma_\mathbf{x}) \\ \mathbf{x}_{t+1} = h(\mathbf{x}_t) + \mathbf{u}_{t+1} \end{cases} \tag{9}$$

où $h(\mathbf{x}_t) = A \left(\psi(x_t^1) \dots \psi(x_t^d) \right)^T$ et les résidus sont homoscédastiques et distribués selon une loi $\mathbf{u}_{t+1} \sim \mathcal{N}(0, \Sigma_\mathbf{u})$. Nous avons choisi d'étudier un bruit de deux natures. Une première configuration correspond à un bruit de covariance multiple de l'identité : $\Sigma_\mathbf{u} = \sigma^2_\mathbf{u} Id$ avec $\sigma_\mathbf{u} > 0$. Le deuxième cas consiste à prendre un bruit de covariance non diagonale. En l'occurrence, $\Sigma_\mathbf{u}$ est ici une matrice de Toeplitz définie par : $\Sigma_{\mathbf{u},pq} =$

$\nu^{|p-q|}$ pour un certain ν dans $(0, 1)$. La fonction $\psi : \mathbb{R} \to \mathbb{R}$ est non linéaire. Typiquement, $\psi(z) = \exp(-\gamma z^2)$. Via le modèle spécifié par (9) et instancié avec les valeurs de paramètres $\sigma^2_{\mathbf{u}} = 1,\ \nu = 0.7$ et $\gamma = 0.1$, nous avons simulé 4 séries temporelles de dimension 10 avec 50 points de temps avec les combinaisons de structures (A) et de covariances ($\Sigma_{\mathbf{u}}$) suivantes :

— pattern « random » et bruit de covariance diagonale
— pattern « random » et bruit de covariance Toeplitz
— pattern « hub » et bruit de covariance diagonale
— pattern « hub » et bruit de covariance Toeplitz



(a) (b)

FIGURE 1 – Motifs de structure générés pour l'évaluation d'OKVAR. (a) Random taille 10 (b) Hub taille 10. Les pixels blancs correspondent aux coefficients non nuls de la matrice d'adjacence A. La matrice B apparaissant dans le noyau K_{dec} est obtenue en projetant A sur le cône des matrices symétriques semi-définies positives.

Sélection de modèle. La sélection des hyperparamètres λ_h et λ_C est réalisée en mettant en œuvre une procédure dite de *validation croisée séquentielle* (cf. section 5.1 dans [5]). L'erreur de validation en un point donné est calculée selon un principe de *fenêtre glissante* de taille fixe en utilisant uniquement les N_w données de la série qui lui sont antérieures.

TABLE 1 – Récapitulatif des modèles OKVAR étudiés. Les poids de la pénalité Ω_{ℓ_1/ℓ_2} sont définis comme suit : $w_\ell = 1 - \exp(-(\ell - 1))$.

	Modèles OKVAR				
	$h^{\text{Ridge}}_{k\cdot Id}$	$h^{\ell_1}_{k\cdot Id}$	$h^{\text{Ridge}}_{\text{dec}}$	$h^{\ell_1}_{\text{dec}}$	$h^{\ell_1/\ell_2}_{\text{dec}}$
Noyau	$k_{\text{Gauss}} \times Id$		K_{dec}, Eq. (3)		
Coût	Eq. (5)				
Ω_C	0	Ω_{ℓ_1}	0	Ω_{ℓ_1}	Ω_{ℓ_1/ℓ_2}

Les résultats de la table 2 indiquent que les modèles à noyaux présentent des performances en prédiction supérieures à celles de VAR(1). D'autre part, les modèles $h_{k\cdot Id}$ qui correspondent en réalité à d modèles à noyaux scalaires gaussiens indépendants obtiennent des performances similaires aux modèles décomposables.

5 Conclusion

Dans cette étude, nous avons présenté les fondements d'une nouvelle famille de modèles autorégressifs vectoriels non paramétriques. Ces modèles reposent sur la théorie des noyaux à valeur opérateur qui offre un cadre élégant pour l'apprentissage de fonctions à valeurs vectorielles. À noyau fixé, nous avons proposé un algorithme proximal permettant d'apprendre les paramètres du modèle, sous des contraintes de parcimonie. Au-delà de la problématique de l'autorégression, tous les outils développés, de par leur généricité, peuvent être appliqués plus largement à des problèmes de prédiction de sorties structurées.

TABLE 2 – Meilleures erreurs moyennes de validation croisée séquentielle des modèles OKVAR, VAR(1) sur les bases de données synthétiques de dimension $d = 10$. Les écarts-types sont donnés entre parenthèses. Taille de la fenêtre utilisée : $N_w = 25$. Les nombres en **gras** sont les plus petites valeurs de chaque colonne.

Structure	Random		Hub	
Bruit	diag.	Toeplitz	diag.	Toeplitz
Modèles				
$h^{\text{Ridge}}_{k\cdot Id}$	1.0673 (0.4946)	1.0218 (0.5475)	1.1048 (0.4597)	1.0783 (0.6670)
$h^{\ell_1}_{k\cdot Id}$	1.0472 (0.4856)	**1.0033** (0.5389)	1.0563 (0.4509)	**1.0310** (0.6772)
$h^{\text{Ridge}}_{\text{dec}}$	1.0499 (0.4834)	1.0441 (0.5806)	1.0527 (0.4387)	1.0762 (0.6785)
$h^{\ell_1}_{\text{dec}}$	**1.0326** (0.4849)	1.0229 (0.5749)	1.0493 (0.4490)	1.0539 (0.6879)
$h^{\ell_1/\ell_2}_{\text{dec}}$	1.0483 (0.4841)	1.0386 (0.5791)	**1.0486** (0.4413)	1.0654 (0.6734)
VAR(1)	1.6324 (0.7431)	1.5403 (1.0474)	1.7039 (0.9698)	2.0463 (1.8402)

Références

[1] M. Alvarez, L. Rosasco, and N. Lawrence. Kernels for vector-valued functions : a review. Technical report, 2011.

[2] A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery problems. In D. Palomar and Y. Eldar, editors, *Convex Optimization in Signal Processing and Communications*, pages 42–88. Cambridge press, 2010.

[3] A. Caponnetto, C. A. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels. *The Journal of Machine Learning Research*, 9 :1615–1646, 2008.

[4] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. In *Journal of Machine Learning Research*, pages 615–637, 2005.

[5] F. Han and H. Liu. A direct estimation of high dimensional stationary vector autoregressions. *arXiv preprint arXiv :1307.0293*, 2013.

[6] N. Lim, F. d'Alché Buc, C. Auliac, and G. Michailidis. Operator-valued kernel-based vector autoregressive models for network inference. *Machine Learning*, pages 1–25, 2014.

[7] C. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17(1) :177–204, 2005.

[8] É. Senkene and A. Tempel'man. Hilbert spaces of operator-valued functions. *Lithuanian Mathematical Journal*, 13(4) :665–670, 1973.

[9] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B*, 68 (1) :49–67, 2006.