Scaling up Vector Autoregressive Models with Operator Random Fourier Features

Romain Brault, Néhémy Lim, Florence d'Alché-Buc

August 22, 2016

Abstract

A nonparametric approach to Vector Autoregressive Modeling consists in working in vector-valued Reproducing Kernel Hilbert Spaces. The main idea is to build vector-valued models (OKVAR) using operator-valued kernels. Similar to scalar-valued kernels, operator-valued kernels enjoy representer theorems and learning algorithms that heavily depends on training data. We present a new approach to scale up OKVAR models... This contribution aims at scaling up non-linear autoregression models based on operator-valued kernel (K) by constructing an explicit feature map function (ORFF) that transforms an input data to a Hilbert space 'embed' in the RKHS induced by K. ORFF are constructed in the spirit of Random Fourier Features introduced by Rahimi and Recht. We show that ORFF competes with VAR on stationary linear time-series in terms of time and accuracy. Moreover ORFF is able to compete with OVK accuracy on non-stationary, non-linear time-series (being better than VAR) while keeping low execution time, comparable to VAR.

1 Introduction

blabla.

2 Models

We compare three models. blabla.

VAR: We fit the model to the data using python statmodels package, available at `http://statsmodels.sourceforge.n`

OVK: We fit the model to the data using python operalib package, available at `https://github.com/RomainBrault/op` The optimization problem is solved using an lbfgs.

ORFF: ORFF aims at approximating kernel $K(x, z) = K_0(x - z)$, by finding an explicit feature map such $\tilde{\Phi}(x)^* \tilde{\Phi}(z) \approx K_0(x - z)$. In the following suppose that $K_0 = k_0(.)A$ is a decomposable kernel on $\mathcal{X} = (\mathbb{R}^d, +)$ and $\mathcal{Y} = \mathbb{R}^p$. Let $A = BB^*$. Then an approximate feature map for K_0 is

$$\tilde{\Phi}^{dec}(x) = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle B^* \\ \sin \langle x, \omega_j \rangle B^* \end{pmatrix}, \ \omega_j \sim \mathcal{F}^{-1}[k_0].$$

Which can also be expressed as a Kronecker product of a scalar feature map with an operator:

$$\tilde{\Phi}^{dec}(x) = \tilde{\phi}(x) \otimes B^*,$$

where,

$$\tilde{\phi}^{dec}(x) = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle \\ \sin \langle x, \omega_j \rangle \end{pmatrix}, \ \omega_j \sim \mathcal{F}^{-1}[k_0]$$

is a scalar valued feature map. In particular if k_0 is a Gaussian kernel of bandwidth σ^2, then $\mathcal{F}^{-1}[k_0] = \mathcal{N}(0, 1/\sigma^2)$. The optimization problem is solved using a mini-batch block coordinate descent. Note that the convergence of the algorithm can be speed-up by preconditioning by the Hessian of the system.

Data: $\mathcal{X}$, $\mathcal{Y}$, K_0, γ_t, λ, T, n, D, b
Result: how to write algorithm with LaTeX2e
Find (ω, x), $B(\omega)$ and $\mu(\omega)$ from K_0;
for $i = 1$ **to** D **do**
  $\theta^1_{i,\cdot} = 0$;
end
for $t = 1$ **to** T **do**
  $\mathcal{A}_t = (\mathcal{X}_t \times \mathcal{Y}_t) \sim \mathbb{P}(x, y)$; `// Sample` n `data from` $\mathcal{X} \times \mathcal{Y}$.
  $f(\mathcal{X}_t) =$ **predict** $(\mathcal{X}_t, \theta^t, K_0)$; `// Make a prediction.`
  $\Omega_i \sim \mu(\omega)$ with seed i, where $i = ((t-1)$ **mod** $D) + 1$; `// Sample` b `features from` $\mu(\omega)$.
  for $\omega \in \Omega_i$ **do**
    $\theta^{t+1}_{i,\omega} = \theta^t_{i,\omega} - \gamma_t \left(\frac{1}{|\mathcal{A}_t|} \sum_{x,y \in \mathcal{A}_t} (\omega, x) B(\omega) l'(f(x), y) + \lambda \theta^t_{i,\omega} \right)$; `// Update the gradient.`
  end
end
return θ^{t+1}

Algorithm 1: Block-coordinate mini-batch SGD.

Data: $\mathcal{X}$, θ, K_0
Find (ω, x), $B(\omega)$ and $\mu(\omega)$ from K_0;
$f = 0$;
for $x \in \mathcal{X}$ **do**
  for $i = 1$ **to** D **do**
    $\Omega_i \sim \mu(\omega)$ with seed i;
    for $\omega \in \Omega_i$ **do**
      $f(x) = f(x) + (\omega, x) B(\omega) \theta_{i,\omega}$;
    end
  end
end
return $f(\mathcal{X})$

Algorithm 2: $f(\mathcal{X}) =$**predict**$(\mathcal{X}, \theta, \mu)$

3 Experiments

3.1 Simulated data

3.1.1 Data generation

A *non-linear multi-time* serie y_t of *dimension* p and *order* one has the form

$$\begin{cases} y_1 \sim \mathcal{N}(0, \Sigma_x) \\ y_t = h(y_{t-1}) + u_t \quad \forall t > 1. \end{cases} \tag{1}$$

Throughout the experiments the residuals considered are homoscedastic and distributed according to a probability measure $u_t \sim \mathcal{N}(0, \Sigma_u)$. We study two different kind of noise: an isotropic with covariance $\Sigma_u = \sigma^2 I_p$ and an anisotropic with Toeplitz structure $\Sigma_{u,ij} = \nu^{|i-j|}$, where ν lives in $(0, 1)$. We generated $N = 10^3$ datapoints and used a sequential cross-validation with time windows $N_t = N/2$ to measure performance of the different models.

3.1.2 Setting 1: Linear model

We first study the behavior of the three method on a linear VAR model (i.e. $h(x) = Ax$). The generated time-series are presented in fig. 1 and fig. 2.

In this setting we do not seed any advantages of OVKs over VAR model. Although OVKs takes order of magnitudes more times to achieve the same performance than OVK, ORFF (the approximation of OVK) is able to challenge VAR in terms of time and accuracy. We fixed $D = 25$ features for ORFF.

model	VAR(1)		ORFF		OVK		Dumb	
noise	White	Toeplitz	White	Toeplitz	White	Toeplitz	White	Toeplitz
SCV-MSE	**0.914979**	**1.091096**	0.919663	1.097183	0.958790	1.410969	1.353183	1.527535
variance	**0.572485**	**1.267880**	0.572936	1.268978	0.591934	1.312243	0.868110	1.501393
time	0.002467(s)	0.004822(s)	**0.000994(s)**	**0.001022(s)**	0.104706(s)	0.289046(s)	0(s)	0(s)

Table 1: Sequential cross-validation MSE on setting 1.

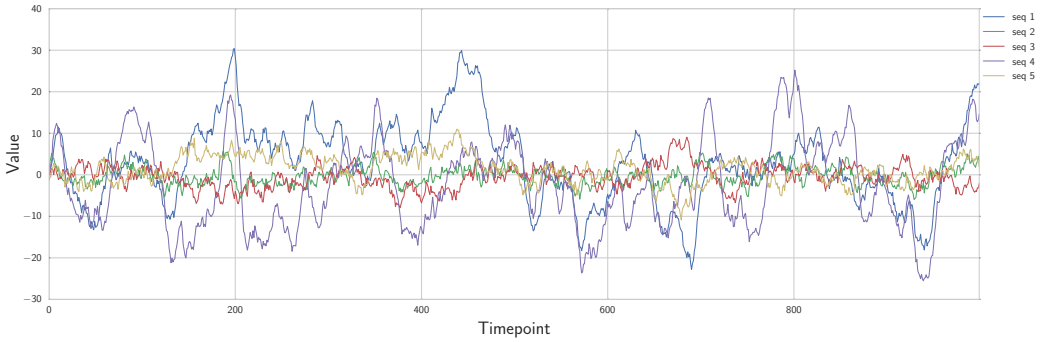

Figure 1: Generated time serie with isotropic noise of variance $\sigma^2 = 0.9$, no non-linearity and random dependency structure with five interactions.

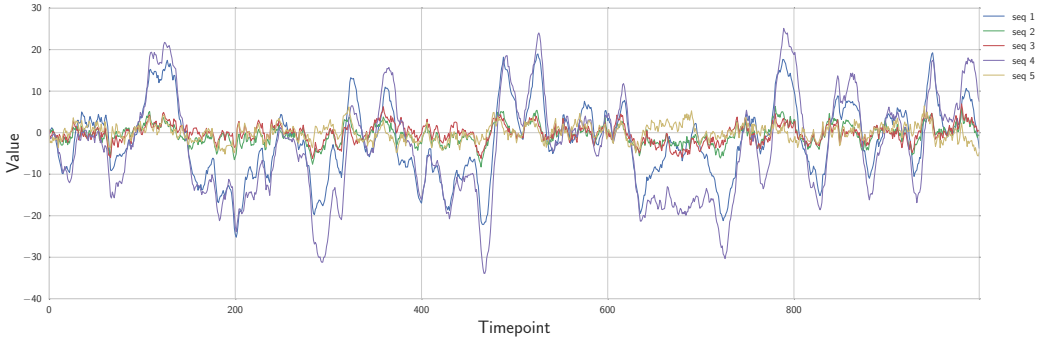

Figure 2: Generated time serie with toeplitz noise of variance $\nu = 0.9$, no non-linearity and random dependency structure with five interactions.

3.1.3 Setting 2: Sine model

We now study the behavior of the three methods on a non-linear VAR model generated by the mean of sin functions (i.e. $h(x) = A\sin(x)$). For this setting the data are generated such that it incorporates a non linear trend f_t. The data y_t are generated according to eq. (2) We chose $f_t = \tilde{\Phi}(t)^*\theta$, where $\theta_{ij} \sim \mathcal{N}(0, 1)$.

$$\begin{cases} x_1 \sim \mathcal{N}(0, \Sigma_x), \\ x_t = h(x_{t-1}) + u_t \quad \forall t > 1, \\ y_t = x_t + f_t \end{cases} \tag{2}$$

The generated time-series are presented in fig. 3 and fig. 4. We fixed $D = 50$ features for ORFF.

model	VAR(1)		ORFF		OVK		Dumb	
noise	White	Toeplitz	White	Toeplitz	White	Toeplitz	White	Toeplitz
SCV-MSE	0.362811	1.883774	0.159656	**1.881269**	**0.101851**	1.883903	0.602598	4.560766
variance	0.222651	1.246095	0.228755	**1.244899**	**0.142791**	1.240014	0.404876	2.429369
time	**0.002998(s)**	0.003791(s)	0.056843(s)	**0.002168(s)**	11.113082(s)	0.079009(s)	0(s)	0(s)

Table 2: Sequential cross-validation MSE on setting 2.

In this setting considering a white noise, non-linear auto-regression with ORFF and OVK has a clear

advantage over VAR(1). ORFF is able to capture the non-linearity in the fraction of time of OVK.

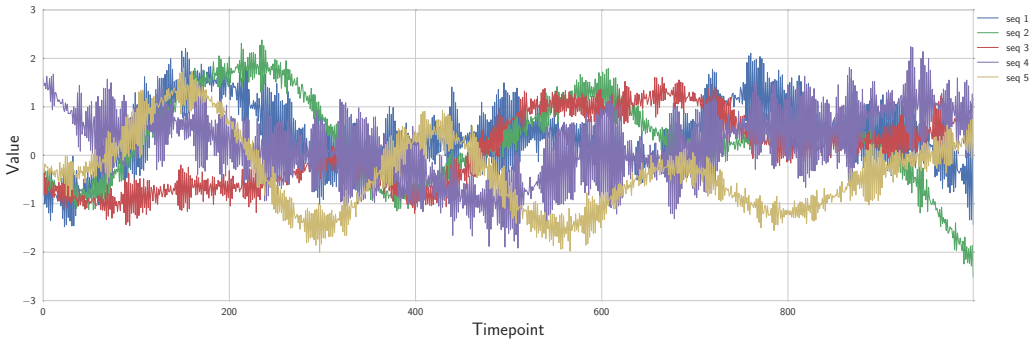

Figure 3: Generated time serie with isotropic noise of variance $\sigma^2 = 0.009$, sine non-linearity ϕ_s and random dependency structure with five interactions.

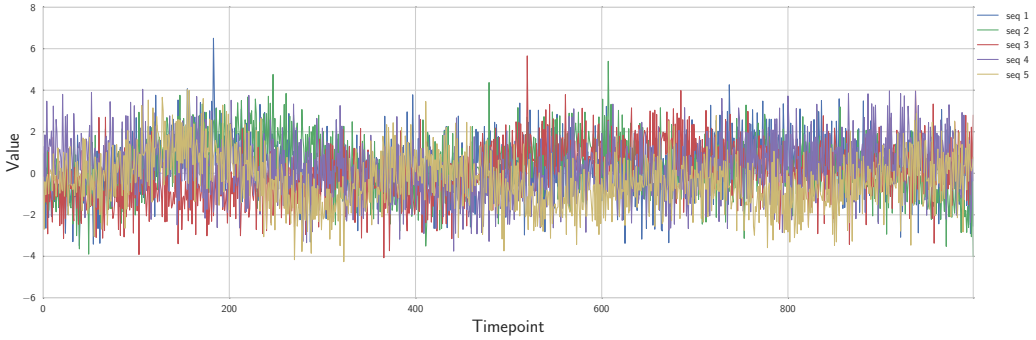

Figure 4: Generated time serie with toeplitz noise of variance $\nu = 0.9$, exponential non-linearity ϕ_e and random dependency structure with five interactions.

3.1.4 Setting 3: Exponential model

Setting 3. follows the same generation model as setting 2 (see eq. (2)). except that the non-linearities are exponential function, i.e. $h(x) = A \exp(-\gamma x^2)$.

model	VAR(1)		ORFF		OVK		Dumb	
noise	White	Toeplitz	White	Toeplitz	White	Toeplitz	White	Toeplitz
SCV-MSE	0.231230	0.514921	**0.001226**	0.252975	0.002951	**0.122719**	1.400892	0.954048
variance	0.248992	0.296253	**0.001086**	0.361397	0.002547	**0.235188**	0.016690	0.228641
time	0.002972(s)	0.003604(s)	**0.002865(s)**	**0.002745(s)**	1.140541(s)	5.369245(s)	0(s)	0(s)

Table 3: Sequential cross-validation MSE on setting 3.

4 Real data

5 Conclusions

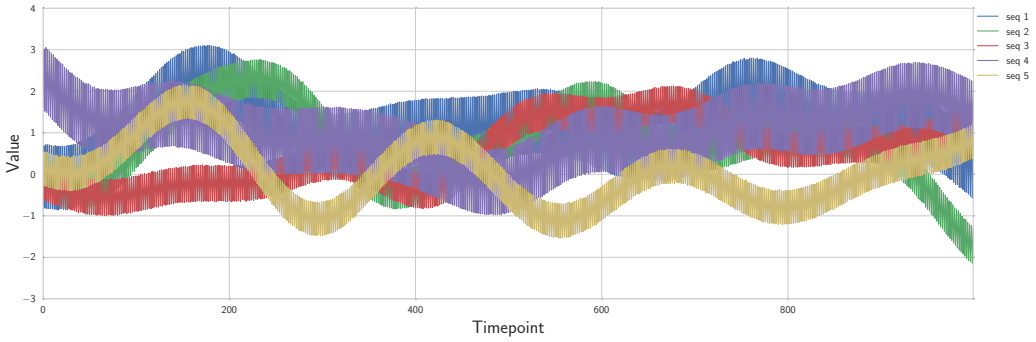

Figure 5: Generated time serie with isotropic noise of variance $\sigma^2 = 0.009$, sine non-linearity ϕ_s and random dependency structure with five interactions.

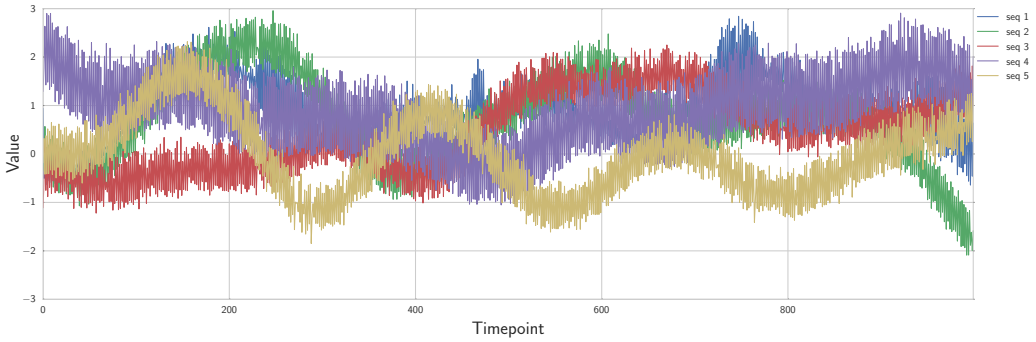

Figure 6: Generated time serie with toeplitz noise of variance $\nu = 0.9$, exponential non-linearity ϕ_e and random dependency structure with five interactions.

6 Supplementary material.