Kernel autoregressive models using Yule-Walker equations

M. Kallas[a,b], P. Honeine[a], C. Francis[b], H. Amoud[c]

[a] *Institut Charles Delaunay (CNRS), LM2S, Université de Technologie de Troyes, France*
[b] *Laboratoire d'analyse des systèmes, Faculté de Génie 1, Université Libanaise, Lebanon*
[c] *Azm Center for Research in Biotechnology and its Applications, Doctoral School for Sciences and Technology, Lebanese University, Lebanon*

Abstract

This paper proposes nonlinear autoregressive (AR) models for time series, within the framework of kernel machines. Two models are investigated. In the first proposed model, the AR model is defined on the mapped samples in the feature space. In order to predict a future sample, this formulation requires to solve a pre-image problem to get back to the input space. We derive an iterative technique to provide a fine-tuned solution to this problem. The second model bypasses the pre-image problem, by defining the AR model with an hybrid model, as a tradeoff considering the computational time and the precision, by comparing it to the iterative, fine-tuned, model. By considering the stationarity assumption, we derive the corresponding Yule-Walker equations for each model, and show the ease of solving these problems. The relevance of the proposed models is studied on several time series, and compared with other well-known models in terms of accuracy and computational complexity.

Keywords: kernel machines, autoregressive model, time series prediction, Yule-Walker equations, pre-image problem

Email addresses: maya.kallas@utt.fr (M. Kallas), paul.honeine@utt.fr (P. Honeine), cfrancis@ul.edu.lb (C. Francis), hassan.amoud@gmail.com (H. Amoud)

1. Introduction

The autoregressive (AR), or linear predictive, model is pervasive in science and technology, with an essential role in the analysis of time series in applications ranging from financial forecasting, to meteorological analysis, to speech processing. For instance to maintain a phone conversation, every cell phone estimates a linear model every 20 milliseconds [1]. The underlying mathematics that govern the AR model are the Yule-Walker equations. The scientific community has made an ever-growing investment to master these equations for the linear prediction [2]. The Yule-Walker equations are the building block of the linear AR model, connecting its parameters to the covariance function of the process. The model parameters are therefore estimated from the covariances of the time series. Forecasting can be considered by applying the resulting predictive model. However, the linearity assumption is often insufficient to explain nonlinear phenomena. A first attempt to derive a nonlinear Yule-Walker like procedure for a specific nonlinear, high-order, model is given in [3]. Nevertheless, up to our knowledge, there is no work that combine the power of the Yule-Walker equations with the proliferating kernel-based methods.

Kernel machines are essentially based on a nonlinear transformation of the data, by using a mapping function from the input space to some feature space, prior to applying a linear procedure in the latter space [4]. Nevertheless, it is not necessary to explicitly define the nonlinear transformation, but implicitly by considering a (positive semi-definite) kernel function. The use of kernel machines has received considerable attention since Vapnik's Support Vector Machines (SVM) [5]. Many nonlinear techniques have been derived, such as the kernel principal component analysis, kernel Fisher discriminant

analysis, and SVM novelty detection, only to name a few. In the same spirit, some kernel-based methods were considered for the analysis and prediction of time series data [6], including the SVM regression and kernel Kalman filter [7].

In this paper, we derive nonlinear prediction models by taking full advantage of the Yule-Walker equations. This leads to the estimation of the model parameters by using lagged expected kernels. It is worth noting that the concept of expected kernels has shown its efficiency in recent research [8, 9]. Two models are under investigation in this paper.

The first model is based on the underlying concept behind kernel machines, namely mapping data from an input space to a feature space. By operating an AR model on the images of the samples, prediction is defined in the feature space. To provide the predicted sample, one needs to get back to the input space, namely to the space of samples. This is the pre-image problem, with a solution (sample in the input space) that has an image as close as possible to the predicted feature (in the feature space). By following recent developments in the resolution of this ill-posed problem [10], we derive an iterative technique to provide a fine-tuned solution to this problem. We propose to bypass the pre-image problem, by deriving another model. In the second model, we propose an hybrid formulation, as a tradeoff considering the computational time and the precision, compared to the iterative, fine-tuned, model.

The rest of the paper is organized as follows: In the next section, we introduce the linear AR model and present the Yule-Walker equations for estimating the model parameters, and give the main idea behind kernel machines in Section 3. The first model is derived in 4, by applying the AR model on the images of the samples, and solving the pre-image problem to interpret the prediction in the input space. Section 5 provides pre-

image-free techniques, by deriving an AR model on the kernel values. Finally, section 7 illustrates the efficiency of the proposed models on several time series data, and provides a comparative study with well-known prediction methods.

2. The Yule-Walker equations of the linear autoregressive model

The linear AR model defines each sample as a linear combination of previous samples. Let $x_1, x_2, \ldots, x_n$ be a time series, the p-order AR model is described by

$$x_i = \sum_{j=1}^{p} \alpha_j \, x_{i-j} + \varepsilon_i, \tag{1}$$

for $i = p+1, \ldots, n$, and where ε_i is the unfitness error, often assumed white Gaussian with zero mean. Figure 1 illustrates the concept of the AR model. The parameters $\alpha_1, \alpha_2, \ldots, \alpha_p$ are directly connected with the covariance function of the process. One can therefore determine these parameters from the autocorrelation function. This is the essence of the Yule-Walker equations, as illustrated here.

[Figure 1 about here.]

Let the data be centered, thus let μ be the expectation of x_i, namely,

$$\mu = \mathbb{E}[x_i],$$

where $\mathbb{E}[\cdot]$ is the expectation[1]. If we apply the expectation on each side of (1), we get that $(1 - \sum_{j=1}^{p} \alpha_j)\mu = \mathbb{E}[\varepsilon_i]$. For any positive lag τ, we can evaluate the autocorrelation function of each time series. Let r be the empirical counterpart of the autocorrelation function of the time series, then $r(\tau) = \sum_{j=1}^{p} \alpha_j \, r(\tau - j)$, for any lag $\tau \geq 1$. Since the

[1] In this paper, all expectations are taken on the index i.

autocorrelation function is even, *i.e.*, $r(-\tau) = r(\tau)$, we obtain the matrix form of the Yule Walker equations

$$\boldsymbol{r} = \boldsymbol{R}\,\boldsymbol{\alpha},$$

where $\boldsymbol{r} = [r(1) \;\; r(2) \;\; \cdots \;\; r(p)]^\top$, $\boldsymbol{\alpha} = [\alpha_1 \;\; \alpha_2 \;\; \cdots \;\; \alpha_p]^\top$, and

$$\boldsymbol{R} = \begin{bmatrix} r(0) & r(1) & \dots & r(p-1) \\ r(1) & r(0) & \dots & r(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \dots & r(0) \end{bmatrix}.$$

Assuming that the $p \times p$ symmetric matrix $\boldsymbol{R}$ is invertible, the coefficients $\boldsymbol{\alpha}$ are estimated by $\boldsymbol{\alpha} = \boldsymbol{R}^{-1}\boldsymbol{r}$. Once the coefficients are estimated, the AR model can be applied to predict future samples, with $x_k = \sum_{j=1}^{p} \alpha_j \, x_{k-j}$.

While this technique is easy to implement, it is not adapted for nonlinear systems. Next, we derive Yule-Walker-like equations for nonlinear models, within the framework of kernel machines. But before, we prepare the ground by briefly describing the main idea behind the kernel machines.

3. Kernel machines

A kernel is a symmetric and continuous function defined by $\kappa \colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ is an input space. If the kernel verifies $\sum_{i,j} \alpha_i \, \alpha_j \, \kappa(x_i, \, x_j) \geq 0$ for all $\alpha_i, \alpha_j \in \mathbb{R}$ and all $x_i, x_j \in \mathcal{X}$, then the kernel is positive semi-definite. The Moore-Aronszajn theorem [11] states that each positive semi-definite kernel defines a unique (up to an isometry) feature space and vice-versa. This feature space $\mathcal{H}$ is obtained using a mapping function

$\Phi\colon \mathcal{X} \mapsto \mathcal{H}$ from the input space, such that

$$\kappa(x_i,\, x_j) = \langle \Phi(x_i),\, \Phi(x_j) \rangle,$$

for any $x_i, x_j \in \mathcal{X}$, where $\langle \cdot\ ,\ \cdot \rangle$ denotes the corresponding inner product in $\mathcal{H}$.

Table 1 summarizes some of the most commonly used kernel functions, grouped into two classes: projective kernels written in terms of dot product $x_i \cdot x_j$ and radial kernels in terms of Euclidean distance $\|x_i - x_j\|$. It is worth noting that some kernels induce infinite-dimensional feature spaces, such as the Gaussian kernel.

[Table 1 about here.]

4. Autoregressive Φ-model

To derive a nonlinear autoregressive model, a straightforward approach consists of applying a nonlinear transformation on the samples, prior to the autoregressive model. This is illustrated in Figure 2, where the samples are mapped from the input space to a feature space using a nonlinear transformation. An AR model is applied to predict a feature in the feature space. Finally, to predict a future sample, one needs to get back to the input space, by solving the so-called pre-image problem. These two stages are derived next.

[Figure 2 about here.]

4.1. Autoregressive model in feature space

Let us start by evaluating the autoregressive model in the feature space. To this end, let $\kappa(\cdot, \cdot)$ be a (reproducing) kernel, defining a nonlinear map $\Phi(\cdot)$ from the input space $\mathcal{X}$ to some feature space $\mathcal{H}$. Thus, each sample x_i is mapped to its corresponding $\Phi(x_i)$.

Therefore, to a time series, given by the samples $x_1, x_2, \ldots, x_n \in \mathcal{X}$, corresponds the samples $\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_n)$ in $\mathcal{H}$. By applying the AR model (1) for the latter, we can write

$$\Phi(x_i) = \sum_{j=1}^{p} \alpha_j \, \Phi(x_{i-j}) + \varepsilon_i^{\Phi}, \tag{2}$$

where these terms are given in the feature space, including $\varepsilon_i^{\Phi} \in \mathcal{H}$ which represents the unfitness of the model. While the samples x_i's are assumed zero-mean in the input space, this is not the case for the $\Phi(x_i)$'s in the feature space.

Let μ_Φ denote the expectation of the latter, namely

$$\mu_\Phi = \mathbb{E}[\Phi(x_i)].$$

On the one hand, by applying the expectation on both sides of (2), we get $(1 - \sum_{j=1}^{p} \alpha_j)\mu_\Phi = \mathbb{E}[\varepsilon_i^{\Phi}]$, where the process is assumed stationary. On the other hand, we have

$$\begin{aligned}
\Phi(x_i) - \mu_\Phi &= \sum_{j=1}^{p} \alpha_j \Phi(x_{i-j}) + \varepsilon_i^{\Phi} - \mu_\Phi \\
&= \sum_{j=1}^{p} \alpha_j \big(\Phi(x_{i-j}) - \mu_\Phi\big) + \varepsilon_i^{\Phi} - \Big(1 - \sum_{j=1}^{p} \alpha_j\Big)\mu_\Phi.
\end{aligned}$$

By combining these results, and by taking the inner product (in the feature space) of both sides of the above equation with $(\Phi(x_{i-\tau}) - \mu_\Phi)$, for some positive lag τ, we get

$$\begin{aligned}
\langle \Phi(x_i) - \mu_\Phi, \, \Phi(x_{i-\tau}) - \mu_\Phi \rangle &= \langle \varepsilon_i^{\Phi} - \mathbb{E}[\varepsilon_i^{\Phi}], \, \Phi(x_{i-\tau}) - \mu_\Phi \rangle \\
&\quad + \sum_{j=1}^{p} \alpha_j \langle \Phi(x_{i-j}) - \mu_\Phi, \, \Phi(x_{i-\tau}) - \mu_\Phi \rangle.
\end{aligned} \tag{3}$$

By analogy with the linear AR case, we assume that the noise ε_i^{Φ} and $\Phi(x_{i-\tau})$ are uncorrelated for every positive lag τ. Therefore, taking the expectations of expression

(3) and assuming a stationarity sequence, we get for any $\tau \geq 1$:

$$\mathbb{E}[\kappa_c(x_i, x_{i-\tau})] = \sum_{j=1}^{p} \alpha_j \, \mathbb{E}[\kappa_c(x_{i-j}, x_{i-\tau})], \tag{4}$$

where $\kappa_c(\cdot, \cdot)$ is the *centered version* of the kernel $\kappa(\cdot, \cdot)$, defined with

$$\kappa_c(x_i, x_j) = \langle \Phi(x_i) - \mu_\Phi, \, \Phi(x_j) - \mu_\Phi \rangle.$$

Finally, we considered all the lag values, and write the expression (4) in matrix form,

$$\boldsymbol{r}_{\kappa_c} = \boldsymbol{R}_{\kappa_c} \, \boldsymbol{\alpha},$$

where

$$\boldsymbol{r}_{\kappa_c} = \Big[\mathbb{E}[\kappa_c(x_i, x_{i-1})] \quad \mathbb{E}[\kappa_c(x_i, x_{i-2})] \; \cdots \; \mathbb{E}[\kappa_c(x_i, x_{i-p})] \Big]^\top,$$

and $\boldsymbol{R}_{\kappa_c}$ is the matrix described by the expected kernels with

$$\boldsymbol{R}_{\kappa_c} = \begin{bmatrix} \mathbb{E}[\kappa_c(x_i, x_i)] & \mathbb{E}[\kappa_c(x_i, x_{i-1})] & \cdots & \mathbb{E}[\kappa_c(x_i, x_{i-p+1})] \\ \mathbb{E}[\kappa_c(x_i, x_{i-1})] & \mathbb{E}[\kappa_c(x_i, x_i)] & \cdots & \mathbb{E}[\kappa_c(x_i, x_{i-p+2})] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{E}[\kappa_c(x_i, x_{i-p+1})] & \mathbb{E}[\kappa_c(x_i, x_{i-p+2})] & \cdots & \mathbb{E}[\kappa_c(x_i, x_i)] \end{bmatrix}.$$

The vector of coefficients $\boldsymbol{\alpha}$ is obtained by inverting the matrix $\boldsymbol{R}_{\kappa_c}$, with

$$\boldsymbol{\alpha} = \boldsymbol{R}_{\kappa_c}^{-1} \boldsymbol{r}_{\kappa_c}.$$

In practice, the expectations are estimated over a set of n available samples (See [12] for a recent review). The centered version of the kernel is evaluated using

$$\kappa_c(x_i, x_j) = \kappa(x_i, x_j) - \frac{1}{n} \sum_{k=1}^{n} \kappa(x_i, x_k) - \frac{1}{n} \sum_{k=1}^{n} \kappa(x_j, x_k) + \frac{1}{n^2} \sum_{k,k'=1}^{n} \kappa(x_{k'}, x_k).$$

4.2. A prediction scheme by solving the pre-image problem

Once the model parameters are determined on a set of n available samples, one can predict by

$$\psi_i = \sum_{j=1}^{p} \alpha_j \, \Phi(x_{i-j}), \tag{5}$$

where the predicted ψ_i lies in the feature space. Still, one is more interested in the predicted sample in the original input space. Thus, we need to map back ψ_i from the feature space to the input space, namely to the so-called pre-image.

The exact pre-image may not exist in general, and even if it exists, it may not be unique. This is the ill-posed *pre-image problem*, where one identifies the best x_i^* in the input space whose image $\Phi(x_i^*)$ is as close as possible to ψ_i. See [10] for a recent review. We propose to solve the optimization problem by minimizing the distance between elements in the feature space, in the same spirit as in [13], namely

$$x_i^* = \arg\min_x \|\psi_i - \Phi(x)\|^2.$$

We propose to use the iterative fixed-point method. By injecting the autoregressive model (5) into the above expression, we obtain the following optimization problem for any x_i:

$$x_i^* = \arg\min_x \left\| \sum_{j=1}^{p} \alpha_j \, \Phi(x_{i-j}) - \Phi(x) \right\|^2.$$

This optimization problem can be written as

$$x_i^* = \arg\min_x J_i(x),$$

where $J_i(x)$ is the cost function defined by

$$J_i(x) = -\sum_{j=1}^{p} \alpha_j \, \kappa(x_{i-j}, x) + \frac{1}{2}\kappa(x, x), \tag{6}$$

where the term independent of x has been removed.

Consider the case of the Radial Basis Functions, with kernels of the form

$$\kappa(x_j, x_{j'}) = f(\|x_j - x_{j'}\|^2). \tag{7}$$

The gradient of the function $\kappa(x_{i-j}, x)$ with respect to x is given by

$$\nabla_x \kappa(x_{i-j}, x) = 2\,(x_{i-j} - x)\, f^{(1)}(\|x_{i-j} - x\|^2),$$

where $f'(z)$ is the first derivative of $f(\cdot)$ w.r.t z. By combining this expression with the gradient of the cost function $J_i(x)$, we get

$$\nabla_x J_i(x) = 2 \sum_{j=1}^{p} \alpha_j\,(x_{i-j} - x)\, f^{(1)}(\|x_{i-j} - x\|^2).$$

Such expression simplifies further for several kernel functions, such as the Gaussian kernel with $f(\zeta) = \exp\left(\frac{-1}{2\sigma^2}\zeta\right)$, thus $f^{(1)}(\zeta) = -\frac{1}{2\sigma^2} f(\zeta)$.

Let us now write the pre-image using a linear combination of the available data, that is $x_i^* = \sum_{j=1}^{p} \delta_j^*\, x_{i-j}$. Many pre-image techniques from the literature [14, 15, 16] have validated this statement. We prove this statement for the radial and projective kernels.

Theorem 1. *Any pre-image x_i^* can be written as a linear combination of the available data, namely*

$$x_i^* = \sum_{j=1}^{p} \delta_j^*\, x_{i-j}$$

for some weights δ_j^*.

Proof. First, we study the class of radial kernels, defined by expression (7). In such case, the term $\partial\kappa(x, x)/\partial x$ vanishes. The gradient at the optimum can be written as

$$\sum_{j=1}^{p} \alpha_j \frac{\partial \kappa(x_{i-j}, x_i^*)}{\partial x_i^*} = 0,$$

with the left-hand-side given as

$$\sum_{j=1}^{p} \alpha_j \frac{\partial \kappa(x_{i-j}, x_i^*)}{\partial x_i^*} = \sum_{j=1}^{p} \alpha_j \frac{\partial f(\|x_{i-j} - x_i^*\|^2)}{\partial(\|x_{i-j} - x_i^*\|^2)}\ 2(x_i^* - x_{i-j}).$$

As a consequence, the final result can be expressed as

$$x_i^* = \sum_{j=1}^{p} \alpha_j \frac{f^{(1)}(\|x_{i-j} - x_i^*\|^2)}{\sum_{k=1}^{p} \alpha_k\, f^{(1)}(\|x_{i-k} - x_i^*\|^2)}\ x_{i-j},$$

thus of the form $x_i^* = \sum_{j=1}^{p} \delta_j^*\, x_{i-j}$.

We now study the projective kernels, of the form $g(x_i \cdot x_j)$. In this case, at the optimum, the gradient is written as

$$\sum_{j=1}^{p} \alpha_j \frac{\partial \kappa(x_{i-j}, x_i^*)}{\partial x_i^*} = \frac{1}{2} \frac{\partial \kappa(x_i^*, x_i^*)}{\partial x_i^*}.$$

We evaluate the right-hand-side and the left-hand-side respectively using the form of the kernel, and we combine both expressions to get

$$x_i^* = \sum_{j=1}^{p} \alpha_j \frac{g^{(1)}(x_{i-j} \cdot x_i^*)}{g^{(1)}(x_i^* \cdot x_i^*)}\ x_{i-j},$$

of the form $x_i^* = \sum_{i=1}^{n} \delta_j^*\, x_{i-j}$. □

5. Autoregressive kernel-based models

The Φ-model proposed above requires to solve the ill-posed pre-image problem, for each predicted sample. In this section, we propose to overcome this drawback. To this end, we investigate the estimation of a function defined on $\mathcal{X}^p$, where the desired output is x_i and the input $\boldsymbol{x}_i$ represents the p previous sample with $\boldsymbol{x}_i = [x_{i-1}\ \ x_{i-2}\ \ \cdots\ \ x_{i-p}]^\top$. Having $\mathcal{X}^p$ as the input space, we consider kernels on $\mathcal{X}^p \times \mathcal{X}^p$. Taking into consideration this formulation, we propose two different nonlinear models, and derive the corresponding Yule-Walker equations.

It is worth noting that, for the Gaussian kernel, this approach corresponds to considering the kernel

$$\kappa(\boldsymbol{x}_{i-j}, \boldsymbol{x}_j) = \exp\Big(- \frac{1}{2\sigma^2} \big\| \boldsymbol{x}_{i-j} - \boldsymbol{x}_i \big\|^2 \Big),$$

where the distance between two vectors $\boldsymbol{x}_{i-j}$ and $\boldsymbol{x}_i$ is given as $\|\boldsymbol{x}_{i-j} - \boldsymbol{x}_i\|^2 = \sum_{k=1}^{p} \big(x_{i-j-k} - x_{i-k}\big)^2$. It is easy to see that this leads to the following expression for the Gaussian kernel

$$\kappa(\boldsymbol{x}_{i-j}, \boldsymbol{x}_i) = \prod_{k=1}^{p} \kappa\big(x_{i-j-k}, x_{i-k}\big),$$

thus connecting the multivariate kernel to the univariate one.

5.1. A hybrid autoregressive model

As opposed to the above method, where we apply the autoregressive model to the Φ-images in the feature space, we consider here an autoregressive model on the kernel values. The proposed model is defined by predicting the sample x_i^*. We investigate the estimation of a function $\varphi(\cdot)$, with $\varphi : \mathcal{X}^p \to \mathcal{X}$. The proposed model is defined by

$$\varphi(\boldsymbol{x}_i) = \sum_{j=1}^{p} \beta_j \, \kappa(\boldsymbol{x}_{i-j}, \boldsymbol{x}_i) \, x_{i-j} + \varepsilon_i, \tag{8}$$

such that $\varphi(\boldsymbol{x}_i) = x_i^*$ is the value of the predicted sample in the input space and $\boldsymbol{x}_i$ represents the p previous samples with $\boldsymbol{x}_i = [x_{i-1}\, x_{i-2} \cdots x_{i-p}]^\top$. See Section 5.2 for the motivation behind this model.

Let $\mu_{\boldsymbol{x}}$ be the expectation of $\varphi(\boldsymbol{x}_i)$, and μ_{xj} be the expectation of $\kappa(\boldsymbol{x}_{i-j}, \boldsymbol{x}_i)\, x_{i-j}$, for any $j = 1, 2, \ldots, p$, namely

$$\mu_{\boldsymbol{x}} = \mathbb{E}[\varphi(\boldsymbol{x}_i)]$$

$$\mu_{xj} = \mathbb{E}[\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-j})\, x_{i-j}].$$

By following the developments given in Section 4, and by analogy with expression (4), we get

$$\mathbb{E}[\langle \varphi(\boldsymbol{x}_i) - \mu_{\boldsymbol{x}},\, \varphi(\boldsymbol{x}_{i-\tau}) - \mu_{\boldsymbol{x}} \rangle] = \sum_{j=1}^{p} \beta_j\, \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-j})\, x_{i-j} - \mu_{xj},\, \varphi(\boldsymbol{x}_{i-\tau}) - \mu_{\boldsymbol{x}} \rangle]. \tag{9}$$

for any lag $\tau = 1, 2, \ldots, p$. By considering all the lag values, we get the matrix form $\boldsymbol{r}_\kappa = \boldsymbol{R}_{\kappa\, x}\, \boldsymbol{\beta}$, where $\boldsymbol{r}_\kappa$ is defined by

$$\boldsymbol{r}_\kappa = \Big[\mathbb{E}[\langle \varphi(\boldsymbol{x}_i) - \mu_{\boldsymbol{x}},\, \varphi(\boldsymbol{x}_{i-1}) - \mu_{\boldsymbol{x}} \rangle] \;\cdots\; \mathbb{E}[\langle \varphi(\boldsymbol{x}_i) - \mu_{\boldsymbol{x}},\, \varphi(\boldsymbol{x}_{i-p}) - \mu_{\boldsymbol{x}} \rangle] \Big]^\top \tag{10}$$

and $\boldsymbol{R}_{\kappa\, x}$ is the matrix described by

$$\boldsymbol{R}_{\kappa\, x} = \begin{bmatrix} \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) x_{i-1} - \mu_{x1},\, \varphi(\boldsymbol{x}_{i-1}) - \mu_{\boldsymbol{x}} \rangle] & \ldots & \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-p}) x_{i-p} - \mu_{xp},\, \varphi(\boldsymbol{x}_{i-1}) - \mu_{\boldsymbol{x}} \rangle] \\ \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) x_{i-1} - \mu_{x1},\, \varphi(\boldsymbol{x}_{i-2}) - \mu_{\boldsymbol{x}} \rangle] & \ldots & \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-p}) x_{i-p} - \mu_{xp},\, \varphi(\boldsymbol{x}_{i-2}) - \mu_{\boldsymbol{x}} \rangle] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) x_{i-1} - \mu_{x1},\, \varphi(\boldsymbol{x}_{i-p}) - \mu_{\boldsymbol{x}} \rangle] & \ldots & \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-p}) x_{i-p} - \mu_{xp},\, \varphi(\boldsymbol{x}_{i-p}) - \mu_{\boldsymbol{x}} \rangle] \end{bmatrix}$$

The vector of coefficients $\boldsymbol{\beta}$ is obtained by inverting the matrix $\boldsymbol{R}_\kappa$, with $\boldsymbol{\beta} = \boldsymbol{R}_{\kappa\, x}^{-1} \boldsymbol{r}_\kappa$.

Once the $\boldsymbol{\beta}$ parameters are evaluated, the hybrid autoregressive model can predict future samples by applying $x_i^* = \varphi(\boldsymbol{x}_i)$, namely

$$x_i^* = \sum_{j=1}^{p} \beta_j\, \kappa(\boldsymbol{x}_{i-j}, \boldsymbol{x}_i)\, x_{i-j}. \tag{11}$$

With this proposed method, we evaluate directly the predicted value without the search for a function in the feature space, where the evaluation is in the input space. To this end, we lose the precision we obtain in the feature space, since we evaluate in the input space. In this case, we do not obtain a more precise value for the prediction. However, we gain in time when applying the hybrid technique since we do not need to solve the pre-image problem in order to obtain the predicted value.

5.2. Connecting the hybrid model with the aforementioned model

The main motivation behind the hybrid model is its connection with the aforementioned model, as illustrated here. We consider the Φ-model, as derived in Section 4. Theorem 1 states that the predicted sample at instant i takes the form

$$x_i^* = \sum_{j=1}^{p} \delta_j^* \, x_{i-j}.$$

By restricting the solution to $\delta_j^* = \beta_j \, \kappa(\boldsymbol{x}_{i-j}, \boldsymbol{x}_i)$, we get the hybrid model defined in (11). More precisely, by taking the result of the Theorem 1 for the Gaussian kernel, the coefficients β_j will be $\beta_j = \alpha_j \frac{1}{\sum_{k=1}^{p} \alpha_k \, f^{(1)}(\|x_{i-k} - x_i^*\|^2)}$.

5.3. Autoregressive model on the kernel values

One may also consider the autoregressive model to the Φ-images in the feature space. Therefore, we consider here an autoregressive model on the kernel values. The proposed model is defined by predicting the sample x_i^*, with

$$\varphi(\boldsymbol{x}_i) = \sum_{j=1}^{p} \beta_j \, \kappa(\boldsymbol{x}_{i-j}, \boldsymbol{x}_i) + \varepsilon_i, \tag{12}$$

where $\boldsymbol{x}_i$ represents the p previous sample with $\boldsymbol{x}_i = [x_{i-1} \;\; x_{i-2} \;\; \cdots \;\; x_{i-p}]^\top$, and $\varphi(\cdot)$ is the function defining the prediction of the future sample such as $\varphi(\boldsymbol{x}_i) = x_i$.

Let μ_j the expectation of $\kappa(\boldsymbol{x}_{i-j}, \boldsymbol{x}_i)$, for each $j = 1, 2, \ldots, p$, namely

$$\mu_j = \mathbb{E}[\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-j})],$$

and $\mu_{\boldsymbol{x}} = \mathbb{E}[\varphi(\boldsymbol{x}_i)]$. By following the developments given in Section 4, and by analogy with expression (4), we get, for any lag $\tau = 1, 2, \ldots, p$

$$\boldsymbol{r}_\kappa = \boldsymbol{R}_\kappa \, \boldsymbol{\beta},$$

where $\boldsymbol{r}_\kappa$ defined by (10) and $\boldsymbol{R}_\kappa$ is the matrix described by the expected kernels with

$$\boldsymbol{R}_\kappa = \begin{bmatrix} \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) - \mu_1,\, \varphi(\boldsymbol{x}_{i-1}) - \mu_{\boldsymbol{x}} \rangle] & \dots & \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-p}) - \mu_p,\, \varphi(\boldsymbol{x}_{i-1}) - \mu_{\boldsymbol{x}} \rangle] \\ \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) - \mu_1,\, \varphi(\boldsymbol{x}_{i-2}) - \mu_{\boldsymbol{x}} \rangle] & \dots & \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-p}) - \mu_p,\, \varphi(\boldsymbol{x}_{i-2}) - \mu_{\boldsymbol{x}} \rangle] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) - \mu_1,\, \varphi(\boldsymbol{x}_{i-p}) - \mu_{\boldsymbol{x}} \rangle] & \dots & \mathbb{E}[\langle \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i-p}) - \mu_p,\, \varphi(\boldsymbol{x}_{i-p}) - \mu_{\boldsymbol{x}} \rangle] \end{bmatrix}.$$

The vector of coefficients $\boldsymbol{\beta}$ is obtained by inverting the matrix $\boldsymbol{R}_\kappa$, with

$$\boldsymbol{\beta} = \boldsymbol{R}_\kappa^{-1} \boldsymbol{r}_\kappa.$$

This model provides a coarse representation of the kernel values and does not require any pre-image solving technique. In practice, this AR model performs poorly. For this reason, it will be discarded from the experimental part of this paper.

6. Study on the complexity for each of the aforementioned techniques

In this section, we study the complexity of the AR methods described in this paper. We consider a set of n samples for the training stage and another set of n samples for the testing stage. For both models, we have to evaluate the centered version of the version, the computational complexity needed to center such kernel remains low, and it can be determined as in iterative/online methods with low computational cost, see Annexe A.3 in [17] for more details.

Let us first start with the AR model in the feature space. In order to evaluate the coefficients, we have to invert a matrix $\boldsymbol{R}_{\kappa\, x}$ and multiply it by the vector $\boldsymbol{r}_\kappa$. The dimension of the matrix is $p \times p$, where p is usually a small. The complexity of the inversion is p^3. Once the parameters are estimated, and the coefficients are evaluated, we have to estimate the pre-image using an iterative technique.

We now consider the hybrid model. In order to evaluate the coefficients, we have to invert a matrix $\boldsymbol{R}_{\kappa x}$ and multiply it by the vector $\boldsymbol{r}_{\kappa}$. The dimension of the matrix is $p \times p$, where p is usually a small. The complexity of the inversion is p^3. Once the parameters are estimated, and the coefficients are evaluated. We can directly predict future samples by applying the equation (11). To this end, we have to evaluate the kernel and multiply it by a vector of the p previous sample. Therefore, the complexity here is $n \times p$.

7. Experiments

In this section, we study the relevance of the different models proposed in this paper. We compare them with the classical linear AR model and nonlinear models such as the multilayer perceptron and the support vector regression. To provide a benchmark study, we considered the same experimental settings as in [7]. Four time series are under investigation: two chaotic time series[2] [18], an electrocardiogram (ECG) signal[3] and an electromyogram (EMG) signal having a neuropathy problem[4].

The *Mackey-Glass* univariate time series models the blood cells production evolution. It is defined by the delay differential equation

$$\frac{dx(t)}{dt} = -0.1\, x(t) + \frac{0.2\, x(t-\tau')}{1 + x(t-\tau')^{10}}$$

which, for values of τ' greater than 16.8, shows some highly nonlinear chaotic behavior. In our case, we set $\tau' = 30$, and denote the time series by MG_{30}.

[2]The time series are available at http://www.bme.ogi.edu/∼ericwan/data.html.

[3]The electrocardiogram signal is taken from the MIT-BIH Normal Sinus Rhythm Database, and is available at http://physionet.org/physiobank/database/nsrdb/

[4]The electromyogram signal is taken from the Examples of Electromyograms, and is available at http://http://physionet.org/physiobank/database/emgdb/

A *Lorenz attractor* is the solution of the system defined by the following differential equations

$$\begin{cases} \frac{dx(t)}{dt} = -ax + ay \\ \frac{dy(t)}{dt} = -xz + rx - y \\ \frac{dz(t)}{dt} = \quad xy - bz \end{cases}$$

For our simulations, we set $a = 10, r = 28$ and $b = 8/3$. This multivariate system is decomposed into the estimation of three models, for x, y, and z, separately.

The ECG signal is ECG recordings of subjects referred to the Arrhythmia Laboratory where subjects included in this database were found to have had no significant arrhythmias. The ECG signals are considered to be stationary, for they are taken in a 1 minute of time.

The EMG signal is EMG recording taken from a clinical test used to assess function of muscles and the nerves that control them. The EMG studies are used to help in the diagnosis and management of disorders such as the muscular dystrophies and neuropathies. Nerve conduction studies that measure how well and how fast the nerves conduct impulses are often performed in conjunction with EMG studies.

Each time series is decomposed into two parts. The first $n = 300$ samples are used in the learning stage, for estimating the optimal value of the order p from $p \in [1, 2, \ldots, 5]$, the coefficients in the AR expansion, and the bandwidth σ of the Gaussian kernel. The next $n = 300$ samples are used to evaluate the relevance of the resulting model, by considering the mean square error (MSE) estimated with

$$\varepsilon_{err} = \frac{1}{n} \sum_{i=n+1}^{2n} \|x_i^* - x_i\|^2,$$

where x_i^* is the predicted value at instant i, and x_i is the true value of the time series at

the same time.

Table 2 gives the mean square error and the computational time measured on an Intel Core 2, with a speed of 2.40 GHz and a random access memory of 1.00 GB, for each of the three proposed nonlinear AR models, as well as the linear model. For the pre-image technique, the stopping criterion is given by a lower bound on the tolerance, set to 10^{-6} , while limiting the maximum number of iterations to 50. It is obvious that the AR Φ-model with pre-image technique presents the best MSE, however such fine-tuning requires significant computational resources. The hybrid model presents a good tradeoff between accuracy and computational complexity, and outperforms the linear AR model. The results obtained for the three time series are illustrated in Figure 3.

The experimental settings are similar to the ones given in [7] for *Lorenz* and MG_{30} time series. This allows us to provide a comparative study with different nonlinear techniques provided in the aforementioned paper, such as multilayer perceptron, support vector regressor, and a kernel Kalman filter. Table 3 shows the MSE evaluated on each time series using different nonlinear prediction technique. As we can see, the kernel AR with pre-image presents the best MSE for the MG_{30}, and the hybrid AR model presents the best MSE for the *Lorenz* attractor time series. An adequate model is contributed to each time series depending on the nature of the data.

[Table 2 about here.]

[Table 3 about here.]

[Figure 3 about here.]

8. Conclusion and future work

In this paper, we proposed to derive Yule-Walker equations for nonlinear autoregressive models. To this end, we combined the simplicity of the AR model with the efficiency of the kernel machines in machine learning. Two models were derived, with a third one that can be used. The first model, the AR Φ-model was based on the mapping defined by a kernel, but requires a pre-image technique to get back to the input space. To circumvent the pre-image problem, we considered an hybrid model. We derived Yule-Walker equations for all these AR models, and showed the ease of estimating the model parameters. The relevance of the proposed techniques were illustrated on three time series, and compared to several well-known nonlinear techniques. As we have concluded, the kernel AR with the pre-image presents the best MSE, however it requires significant computational resources. The hybrid model represents a good compromise between efficiency and computational cost, and outrun the linear AR model.

As for future work, we are considering the estimation of the optimal order, by adapting several criteria such as the Akaike Information Criterion, the Bayesian Information Criterion and the partial autocorrelation function. Also, we intend to apply other techniques to estimate the model parameters, such as the Levinson-Durbin method. Finally, we will extend the use of the kernel machines for the Autoregressive Moving Average model.

References

[1] T. Dutoit, N. Moreau, P. Kroon, Eds. T. Dutoit, F. Marques, Applied Signal Processing, Springer, 2009, Ch. How is speech processed in a cell phone conversation?

[2] M. Chevalier, Y. Grenier, Autoregressive models with time-dependent log area ratios, in: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85., Vol. 10, 1985, pp. 1049 – 1052. doi:10.1109/ICASSP.1985.1168137.

[3] R. Cusani, E. Baccarelli, Parameter identification of frequency-selective noisy fast-fading rayleigh digital channels via nonlinear yule-walker-like equations, in: Proc. of European Conference on Signal Processing (EUSIPCO), Eurasip, 1996.

[4] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.

[5] V. N. Vapnik, Statistical Learning Theory, Wiley-Interscience, 1998.

[6] C. Richard, J. C. M. Bermudez, P. Honeine, Online prediction of time series data with kernels, IEEE trans. on Signal Processing 57 (3) (2009) 1058–1067.

[7] L. Ralaivola, F. D'alche-Buc, Time series filtering, smoothing and learning using the kernel kalman filter, in: Proc. IEEE International Joint Conference on Neural Networks, Vol. 3, 2005, pp. 1449–1454.

[8] H. S. Anderson, M. R. Gupta, E. Swanson, K. Jamieson, Channel-robust classifiers, IEEE Transactions on Signal Processing 59 (4) (2011) 1421–1434.

[9] H. S. Anderson, M. R. Gupta, Expected kernel for missing features in support vector machines, in: IEEE Workshop on Statistical Signal Processing, Nice, France, 2011.

[10] P. Honeine, C. Richard, The pre-image problem in kernel-based machine learning, IEEE Signal Processing Magazine, special issue on "dimensionality reduction via subspace and manifold learnin" 28 (2).

[11] N. Aronszajn, Theory of reproducing kernels, Trans. Amer. Math. Soc. 68 (1950) 337–404. doi:10.2307/1990404.

[12] C. Cortes, M. Mohri, A. Rostamizadeh, Algorithms for learning kernels based on centered alignment, Journal of Machine Learning Research 13 (2012) 795–828.

[13] M. Kallas, P. Honeine, C. Richard, H. Amoud, C. Francis, Nonlinear feature extraction using kernel principal component analysis with non-negative pre-image, in: Proc. 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Buenos Aires, Argentina, 2010.

[14] J. T. Kwok, I. W. Tsang, The pre-image problem in kernel methods, in: T. Fawcett, N. Mishra

(Eds.), ICML, AAAI Press, 2003, pp. 408–415.

[15] P. Honeine, C. Richard, Solving the pre-image problem in kernel machines: a direct method, in: Proc. IEEE Workshop on Machine Learning for Signal Processing (MLSP), Grenoble, France, 2009.

[16] W. Zheng, J. Lai, P. C. Yuen, Penalized preimage learning in kernel principal component analysis, IEEE Transaction Neural Networks 21 (2010) 551–570.

[17] K. I. Kim, M. O. Franz, B. Schlkopf, Kernel Hebbian algorithm for iterative kernel principal component analysis, Technical Report in: Max Planck Institute for Biological Cybernetics, 2003.

[18] S. Mukherjee, E. Osuna, F. Girosi, Nonlinear prediction of chaotic time series using support vector machines, in: IEEE Workshop on Neural Networks for Signal Processing VII, IEEE Press, 1997, pp. 511–519.

List of Figures

1 Illustration of the AR model, where the x_i is defined by a linear combination of the p previous samples x_{i-k}'s, with weight parameters $\alpha_1, \alpha_2, \ldots, \alpha_p$. 23

2 Illustration of the AR Φ-model: the samples are mapped from $\mathcal{X}$ to $\mathcal{H}$, where the AR model is applied. Once ψ_i is estimated, one needs to map it back to the input space $\mathcal{X}$, in order to provide the prediction x_i^*. 24

3 Visualization of the results obtained when predicting the four time series using the linear AR model (first column), the AR Φ-model with pre-image (second column), and the hybrid AR model (last column). The predicted time series is given in red dash-point (-·) lines, while the blue lines define the original data. 25

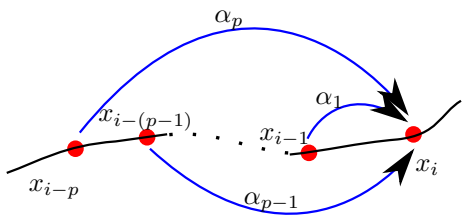

Figure 1: Illustration of the AR model, where the x_i is defined by a linear combination of the p previous samples x_{i-k}'s, with weight parameters $\alpha_1, \alpha_2, \ldots, \alpha_p$.

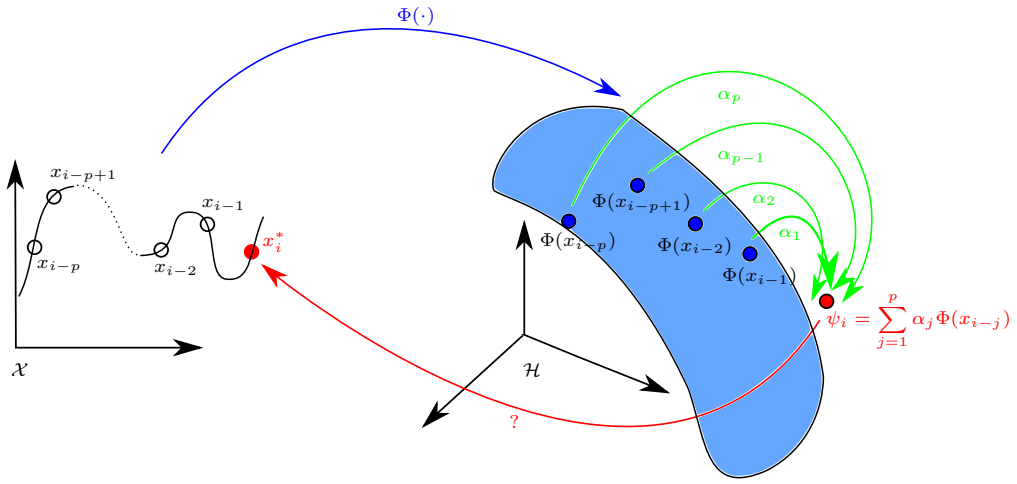

Figure 2: Illustration of the AR Φ-model: the samples are mapped from $\mathcal{X}$ to $\mathcal{H}$, where the AR model is applied. Once ψ_i is estimated, one needs to map it back to the input space $\mathcal{X}$, in order to provide the prediction x_i^*.

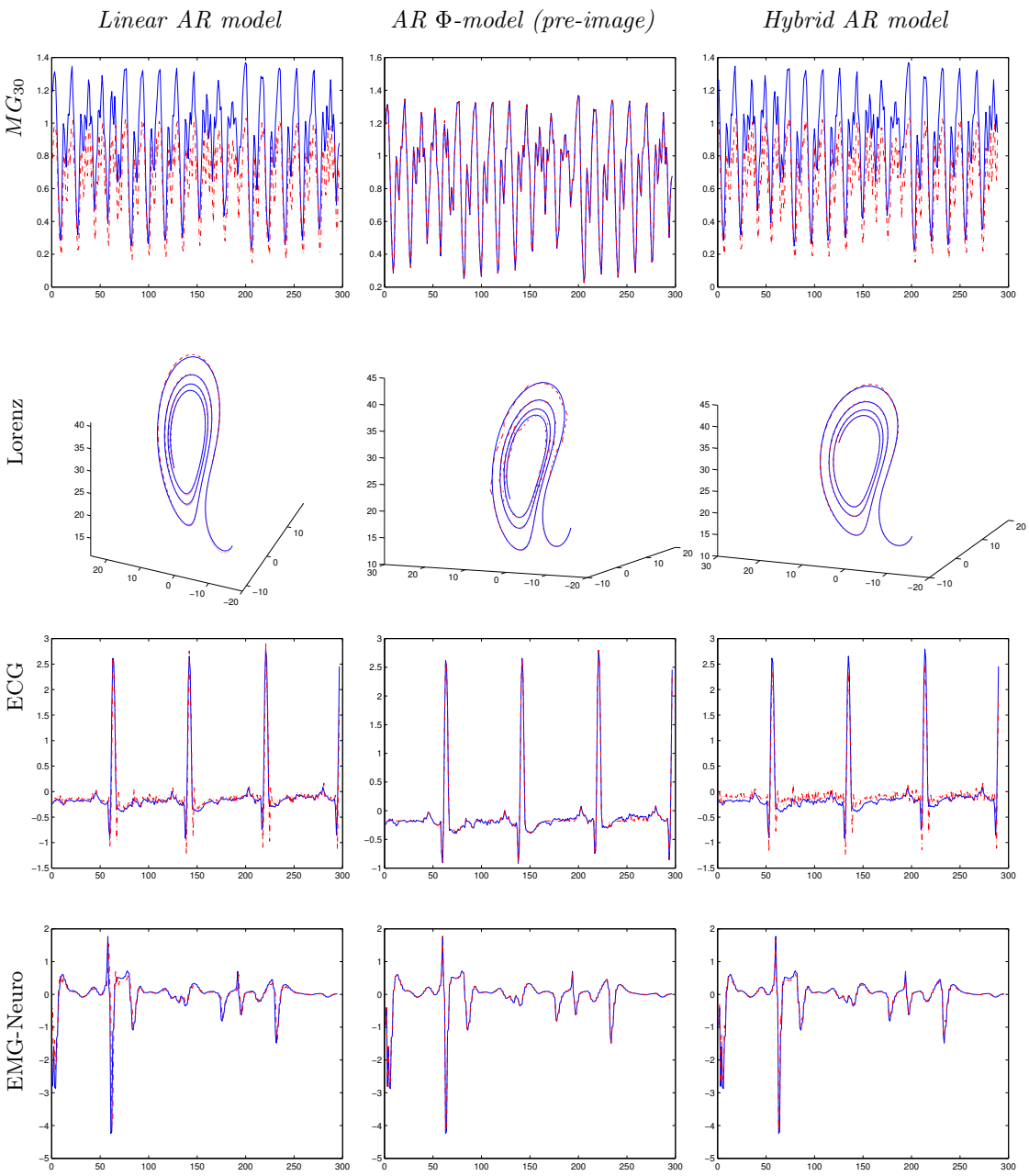

Figure 3: Visualization of the results obtained when predicting the four time series using the linear AR model (first column), the AR Φ-model with pre-image (second column), and the hybrid AR model (last column). The predicted time series is given in red dash-point (-·) lines, while the blue lines define the original data.

List of Tables

1 Commonly used reproducing kernels in machine learning, with parameters $c, \sigma > 0$, and $p \in \mathbb{N}_+$ 27
2 Estimated computational time and mean square error (MSE) between the predicted values and the original ones. 28
3 The mean square error values for different nonlinear prediction techniques compared to our proposed methods. 29

	Type	General form
Projective	Monomial	$\kappa_m(x_i, x_j) = (x_i \cdot x_j)^p$
	Polynomial	$\kappa_p(x_i, x_j) = (c + x_i \cdot x_j)^p$
	Exponential	$\kappa_E(x_i, x_j) = \exp(\frac{1}{\sigma}(x_i \cdot x_j))$
	Sigmoid	$\kappa_S(x_i, x_j) = \tanh(c\,(x_i \cdot x_j) + \sigma)$
Radial	Laplacian	$\kappa_L(x_i, x_j) = \exp(\frac{-1}{\sigma}\|x_i - x_j\|)$
	Gaussian	$\kappa_G(x_i, x_j) = \exp(\frac{-1}{2\sigma^2}\|x_i - x_j\|^2)$
	Multiquadratic	$\kappa_{MQ}(x_i, x_j) = \sqrt{\|x_i - x_j\|^2 + c}$
	Rational	$\kappa_R(x_i, x_j) = 1 - \frac{\|x_i - x_j\|^2}{\|x_i - x_j\|^2 + \sigma}$

Table 1: Commonly used reproducing kernels in machine learning, with parameters $c, \sigma > 0$, and $p \in \mathbb{N}_+$

		MG_{30}	*Lorenz*	*ECG*	*EMG-Neuro*
Linear AR model	Time (s)	0.0107	0.0539	0.0126	0.0133
(with (1))	MSE	0.0655	0.2907	0.0332	0.1397
AR Φ-model (pre-image)	Time (s)	5.3201	10.5241	7.8921	9.0315
(with (4))	MSE	$1 \cdot 10^{-5}$	0.1498	$6 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
Hybrid AR model	Time (s)	0.0861	0.6091	0.1834	0.1280
(with (9))	MSE	0.0623	0.0213	0.0290	0.0061

Table 2: Estimated computational time and mean square error (MSE) between the predicted values and the original ones.

	MG_{30}	*Lorenz*
Multilayer perceptron	0.0461	0.2837
Support vector regression	0.0313	0.1811
Nonlinear Kalman filter	0.0307	0.3133
AR Φ-model (pre-image)	$1 \cdot 10^{-5}$	0.1498
Hybrid AR model	0.0623	0.0213

Table 3: The mean square error values for different nonlinear prediction techniques compared to our proposed methods.