DICTIONARY ADAPTATION FOR ONLINE PREDICTION OF TIME SERIES DATA WITH KERNELS

Chafic Saidé † *Régis Lengellé* † *Paul Honeine* † *Cédric Richard* ‡ *Roger Achkar* ⋆

† Institut Charles Delaunay (UMR CNRS 6279), Université de technologie de Troyes, France
‡ Laboratoire H. Fizeau (UMR CNRS 6525), Université de Nice Sophia-Antipolis, France
⋆ American University of Science and Technology, Beirut, Lebanon

ABSTRACT

During the last few years, kernel methods have been very useful to solve nonlinear identification problems. The main drawback of these methods resides in the fact that the number of elements of the kernel development, i.e., the size of the dictionary, increases with the number of input data, making the solution not suitable for online problems especially time series applications. Recently, Richard, Bermudez and Honeine investigated a method where the size of the dictionary is controlled by a coherence criterion. In this paper, we extend this method by adjusting the dictionary elements in order to reduce the residual error and/or the average size of the dictionary. The proposed method is implemented for time series prediction using the kernel-based affine projection algorithm.

***Index Terms*—** Nonlinear adaptive filters, machine learning, nonlinear systems, kernel methods.

1. INTRODUCTION

Nonlinear models represent a challenge in many practical situations, for which numerous methods have been considered such as neural networks [1] and series expansion methods [2, 3]. Function approximation methods based on reproducing kernel Hilbert spaces (RKHS) are of great importance in kernel-based regression methods such as support vector regression [4, 5]. Computational requirements for kernel-based methods depend on matrices which size increases with the number of observations. This characteristic makes them unsuitable for online applications. Several methods have been proposed to overcome these calculation costs for online applications. The main idea consists in introducing a new sample to the model if it contributes significantly in reducing the approximation error and, if necessary, removing the element which contributes the least. In [6, 7], the authors proposed a sparsification rule based on the orthogonal projections, while in [8] the approximate linear dependence criterion was considered.

The computational cost is further reduced in [9, 10], where the authors introduced the dictionary[1] *coherence* criterion, where the inclusion of a new input data into the dictionary is performed if the dictionary still has a small coherence. They demonstrated that the number of elements in the dictionary remains finite with time. The coherence criterion was coupled with Kernel Affine Projection Algorithm (KAPA) and Kernel Normalized Least Mean Squares (KNLMS) as particular cases.

In all above methods, each element injected into the dictionary remains permanently unchanged, even if the non stationarity makes it later useless for estimating the solution. This is the reason why adaptation of the dictionary appears necessary to obtain a better accuracy and/or a smaller size of the dictionary. In this paper, we study the adjustment of the dictionary elements using an adaptive algorithm to make it more efficient in minimizing the resulting approximation error. We make use of the coherence criterion, which gives us the possibility to reduce, in average, the size of the dictionary. To illustrate the efficiency of the proposed method, we present several experiments using a well known benchmark.

2. BRIEF REVIEW OF THE MODEL REDUCTION METHOD USING THE COHERENCE CRITERION

Consider an online prediction problem and let $\boldsymbol{u}_n \in \mathcal{U}$ the input data vector at time step n and $d_n \in \mathbb{R}$ the corresponding desired output of the model. Let $k : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ be a kernel and $\mathcal{H}$ is the RKHS associated with it. Due to the representer theorem [11, 12], the predicted model can be written as follows:

$$\psi_n(\cdot) = \sum_{j=1}^{m} \alpha_{n,j}\, \kappa(\cdot, \boldsymbol{u}_{w_j}) \tag{1}$$

where the coefficients $\alpha_{n,j} \in \mathbb{R}$ and $\psi_n(\cdot)$ is a real-valued function in the Hilbert space associated with the kernel function κ. The $\kappa(\cdot, \boldsymbol{u}_{w_1}), \ldots, \kappa(\cdot, \boldsymbol{u}_{w_m})$ form a m-elements subset called the dictionary $\mathcal{D}_n$ and $m \ll n$ is the model order by analogy with linear transverse filters. The response of the model to $\boldsymbol{u}_n$ at time n is:

$$\psi_n(\boldsymbol{u}_n) = \sum_{j=1}^{m} \alpha_{n,j}\, \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_j})$$

The main problem in kernel online prediction is the unknown order of the model which increases with time and hence, the necessity to control it at each time step. The coherence criterion is of a great importance to overcome this problem and it is widely used to characterize a dictionary in linear sparse approximation techniques [13].

In [10], the authors proposed to define this parameter for kernel-based models by:

$$\mu = \max_{i \neq j} |\kappa(\boldsymbol{u}_{w_i}, \boldsymbol{u}_{w_j})|$$

where κ is a unit-norm[2] kernel, i.e., $\kappa(\boldsymbol{u}_k, \boldsymbol{u}_k) = 1$ for all $\boldsymbol{u}_k$. The parameter μ is the largest absolute value of the off-diagonal entries in the Gram matrix and reflects the largest cross-correlation of the

[1] The term dictionary stands for a set of input vectors (or their corresponding kernel functions in the RKHS) used to estimate the nonlinear model.

[2] Otherwise, replace $\kappa(\cdot, \boldsymbol{u}_k)$ with $\kappa(\cdot, \boldsymbol{u}_k)/\sqrt{\kappa(\boldsymbol{u}_k, \boldsymbol{u}_k)}$.

elements of the dictionary. The dictionary is said to be μ-coherent. Note that $\mu = 0$ for an orthogonal basis.

In order to derive a model of the form (1) with the coherence parameter, at each time step n, a candidate function $\kappa(\cdot, \boldsymbol{u}_n)$ is introduced into the dictionary if the following condition is satisfied:

$$\max_{\kappa(\cdot, \boldsymbol{u}_{w_j}) \in \mathcal{D}_n} |\kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_j})| \leq \mu_0 \tag{2}$$

where $\mu_0 \in [0, 1[$ is a threshold parameter determining the level of sparsity and the coherence of the dictionary. A very important consequence of the use of the coherence criterion is that the dimension m of the dictionary remains finite as n goes to infinity. Obviously, m increases with μ_0.

3. THE KERNEL AFFINE PROJECTION ALGORITHM

Let $\mathcal{D}_n$ be a μ_0-coherent dictionary at time step n ($\mathcal{D}_n$ satisfies (2)), and m be its order. In (1), the optimal solution vector $\boldsymbol{\alpha}_n = (\alpha_{n,1} \cdots \alpha_{n,m})^t$ is obtained in accordance with the least-squares problem

$$\min_{\boldsymbol{\alpha}_n} = \|\boldsymbol{d}_n - \boldsymbol{H}_n \boldsymbol{\alpha}_n\|^2, \tag{3}$$

where $\mathbf{H}_n$ is a $p \times m$ matrix whose (i, j)-th element is $\kappa(\boldsymbol{u}_{n-i+1}, \boldsymbol{u}_{w_j})$. Let p be the number of inputs/outputs used in (3). This means that, at each time step n, only the p most recent inputs $\{\boldsymbol{u}_n, ..., \boldsymbol{u}_{n-p+1}\}$ and observations $\boldsymbol{d}_n = (d_n \cdots d_{n-p+1})^t$ are considered [14, 15, 16]. When a new input data vector $\boldsymbol{u}_{n+1}$ is fed to the model, one of the following two cases occurs:

- *First case*: $\max_{j=1,...,m} |\kappa(\boldsymbol{u}_{n+1}, \boldsymbol{u}_{w_j})| > \mu_0$
  $\kappa(\cdot, \boldsymbol{u}_{n+1})$ is not introduced into the dictionary $\mathcal{D}_n$ and the solution vector $\boldsymbol{\alpha}_{n+1}$ is updated as follows:

  $$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n + \eta \mathbf{H}^t_{n+1}(\epsilon \mathbf{I} + \mathbf{H}_{n+1}\mathbf{H}^t_{n+1})^{-1}(\boldsymbol{d}_{n+1} - \mathbf{H}_{n+1}\boldsymbol{\alpha}_n)$$

  where η is a step-size control parameter and $\epsilon\mathbf{I}$ is a regularization factor.

- *Second case*: $\max_{j=1,...,m} |\kappa(\boldsymbol{u}_{n+1}, \boldsymbol{u}_{w_j})| \leq \mu_0$
  $\kappa(\cdot, \boldsymbol{u}_{n+1})$ is introduced into the dictionary. Thus, $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{\kappa(\cdot, \boldsymbol{u}_{w_{m+1}})\}$ with $\boldsymbol{u}_{w_{m+1}} = \boldsymbol{u}_{n+1}$. The solution vector $\boldsymbol{\alpha}_{n+1}$ is updated according to:

  $$\boldsymbol{\alpha}_{n+1} = \begin{bmatrix} \boldsymbol{\alpha}_n \\ 0 \end{bmatrix} + \eta \mathbf{H}^t_{n+1}(\epsilon \mathbf{I} + \mathbf{H}_{n+1}\mathbf{H}^t_{n+1})^{-1} \times \left(\boldsymbol{d}_{n+1} - \mathbf{H}_{n+1} \begin{bmatrix} \boldsymbol{\alpha}_n \\ 0 \end{bmatrix} \right)$$

These recursions define the Kernel Affine Projection Algorithm (KAPA). See [10] for more details.

4. DICTIONARY ADJUSTMENT

In this section, we describe our approach to adjust the elements of the dictionary $\mathcal{D}_n$ to obtain $\mathcal{D}^A_n$ using an adaptive method, by substituting each $\boldsymbol{u}_{w_k}$ with an appropriate $\boldsymbol{u}^A_{w_k}$, for $k = 1...m$. The objective is to minimize the quadratic approximation error e_n^2:

$$e_n = d_n - \psi_n(\boldsymbol{u}_n) = d_n - \sum_{j=1}^{m} \alpha_{n,j}\, \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_j}).$$

Since the coherence criterion (2) is satisfied at each time instant, before any adaptation, for all pairs of elements in the dictionary, this constraint must remain satisfied after the adjustment, leading to

$$\max_{i \neq j} |\kappa(\boldsymbol{u}^A_{w_i}, \boldsymbol{u}^A_{w_j})| \leq \mu_0. \tag{4}$$

The i-th element of the dictionary is adapted using a perturbation in the opposite direction of the gradient of the instantaneous quadratic error with respect to $\boldsymbol{u}_{w_i}$, as follows:

$$\boldsymbol{u}^A_{w_i} = \boldsymbol{u}_{w_i} - \nu_n \mathbf{g}_{w_i} \quad \forall i = 1...m \tag{5}$$

where

$$\mathbf{g}_{w_i} = -2e_n\, \alpha_{n,i}\, \nabla_{\boldsymbol{u}_{w_i}} \kappa(\boldsymbol{u}_n, \boldsymbol{u}_{w_i}) \tag{6}$$

is the gradient of the instantaneous quadratic error and ν_n represents the step size used to adjust all the elements of the dictionary. Then, for any pair of dictionary elements, we obtain

$$\boldsymbol{u}^A_{w_i} - \boldsymbol{u}^A_{w_j} = \delta\boldsymbol{u} - \nu_n \delta\mathbf{g} \quad \forall i, j = 1...m$$

where $\delta\boldsymbol{u} = \boldsymbol{u}_{w_i} - \boldsymbol{u}_{w_j}$ and $\delta\mathbf{g} = \mathbf{g}_{w_i} - \mathbf{g}_{w_j}$. Under the coherence constraint, ν_n cannot be chosen arbitrarily. The problem is to determine an appropriate ν_n at each time step n in order to adapt the dictionary.

The iterative approach to find the best step size is explored in the case of Radial Basis Functions, of the form

$$\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = f(\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2), \tag{7}$$

where $f \in \mathcal{C}^\infty$. A sufficient condition for this function to be a valid positive-definite kernel is the complete monotonicity of the function f [17], i.e.,

$$(-1)^k f^{(k)}(r) \geq 0, \forall r \geq 0 \tag{8}$$

where $f^{(k)}(\cdot)$ denotes the k-th derivative of $f(\cdot)$. From (7), we get

$$\nabla_{\boldsymbol{u}_j} \kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = -2(\boldsymbol{u}_i - \boldsymbol{u}_j)\, f^{(1)}(\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2)$$

and the coherence condition (4) leads to

$$f(\|\delta\boldsymbol{u} - \nu_n\, \delta\mathbf{g}\|^2) \leq \mu_0. \tag{9}$$

It is possible to construct a local model of the kernel function, by approximating it with a Taylor series around $\nu_n \sim 0$:

$$f(\|\delta\boldsymbol{u} - \nu_n \delta\mathbf{g}\|^2) = f(\|\delta\boldsymbol{u}\|^2) - 2\nu_n(\delta\boldsymbol{u}^t\delta\mathbf{g} - \nu_n\|\delta\mathbf{g}\|^2) f^{(1)}(\|\delta\boldsymbol{u}\|^2) + \mathcal{O}(\nu_n)$$

Using this approximation, condition (9) becomes

$$-\left(2\|\delta\mathbf{g}\|^2\nu_n^2 - 2\nu_n\delta\boldsymbol{u}^t\delta\mathbf{g}\right) f^{(1)}(\|\delta\boldsymbol{u}\|^2) + \mu_0 - f(\|\delta\boldsymbol{u}\|^2) \geq 0. \tag{10}$$

The discriminant of this quadratic inequality in ν_n is:

$$\Delta = \left(\delta\boldsymbol{u}^t\delta\mathbf{g}\, f^{(1)}(\|\delta\boldsymbol{u}\|^2)\right)^2 + 2\|\delta\mathbf{g}\|^2 f^{(1)}(\|\delta\boldsymbol{u}\|^2)(\mu_0 - f(\|\delta\boldsymbol{u}\|^2)$$

If $\Delta < 0$, there is no constraint on the value of ν_n; otherwise, if $\Delta \geq 0$, the boundary points of (10) will be $\nu_{i,j-}$ and $\nu_{i,j+}$ as follows:

$$\nu_{i,j\pm} = \frac{-\delta\boldsymbol{u}^t\delta\mathbf{g}\, f^{(1)}(\|\delta\boldsymbol{u}\|^2) \pm \sqrt{\Delta}}{-\|\delta\mathbf{g}\|^2 f^{(1)}(\|\delta\boldsymbol{u}\|^2)}$$

Since the quadratic expression in (10) must be positive, the interval of possible values for ν_n is $]-\infty, \nu_{i,j-}] \cup [\nu_{i,j+}, +\infty[$

$$\begin{array}{ccccc} & \nu_{i,j-} & & \nu_{i,j+} & \\ \hline + & | & - & | & + \end{array}$$

Obviously, for each pair of dictionary elements $(\boldsymbol{u}_{w_i}, \boldsymbol{u}_{w_j})$, $\nu_n = 0$ always belongs to the feasible domain, since in this case there is no adaptation of the dictionary, and the last was μ_0-coherent. Because the constant term in (10) is positive, $\nu_{i,j-}$ and $\nu_{i,j+}$ have the same sign[3].

The interpretation of the two bounds for ν_n is straightforward. When adjusting any two elements of the dictionary, each one according to the gradient of the quadratic error with respect to this element, the bounds $(\nu_{i,j-}, \nu_{i,j+})$ must be satisfied to avoid any overlap of the influence regions of the two considered elements, and thus violation of the coherence constraint (4). (see Fig. 1).

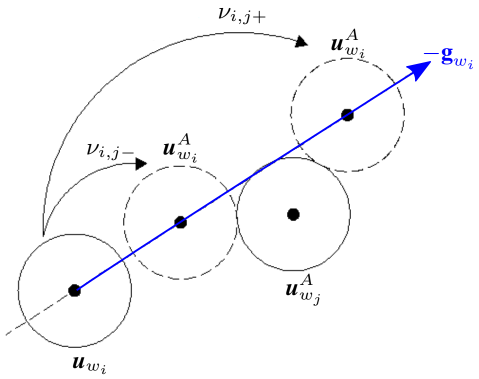

Fig. 1. A 2D illustration showing the constraint of choosing $\nu_n \leq \nu_{i,j-}$ or $\nu_n \geq \nu_{i,j+}$ to avoid the overlap of the influence regions of $\boldsymbol{u}^A_{w_i}$ and $\boldsymbol{u}^A_{w_j}$.

We now propose a heuristic for selecting ν_n. Other heuristics could be considered, but our main objective is to illustrate the efficiency of dictionary adaptation. We initially select a reference step size $\nu_0 > 0$, as commonly done for adaptive algorithms with a fixed step size. By considering all the $(\nu_{i,j-}, \nu_{i,j+})$ pairs, ν_n is selected as follows:

- if $\max\limits_{i,j} \nu_{i,j+} \leq 0 \Rightarrow \nu_n = \nu_0$;
- if $0 \leq \min\limits_{i,j} \nu_{i,j-} \leq \nu_0 \Rightarrow \nu_n = \min\limits_{i,j} \nu_{i,j-}$;
- if $0 \leq \nu_0 \leq \min\limits_{i,j} \nu_{i,j-} \Rightarrow \nu_n = \nu_0$;
- if $0 \leq \min\limits_{i,j}(\nu_{i,j-})^+ \leq \nu_0 \Rightarrow \nu_n = \min\limits_{i,j}(\nu_{i,j-})^+$;
- if $0 \leq \nu_0 \leq \min\limits_{i,j}(\nu_{i,j-})^+ \Rightarrow \nu_n = \nu_0$.

In these expressions, $(\nu_{i,j-})^+$ indicates all the positive values of $\nu_{i,j-}$'s. Note that ν_0 must be selected relatively small. If ν_0 is too large, the elements of the dictionary can be spread over a non useful region of the input space, inducing an increase of the size of the dictionary without reducing the approximation error.

5. EXPERIMENTATIONS

In our experimentation, we used the same benchmark and the same parameter settings as in [10]. It consists of a one step prediction of the following discrete-time nonlinear dynamical system:

$$\begin{cases} v_n = 1.1 \exp(-|v_{n-1}|) + u_n \\ d_n = v_n^2 \end{cases}$$

where u_n and d_n are the input and the desired output, respectively. The data were generated from the initial condition v_0 =0.5. The input was sampled from a zero-mean Gaussian distribution with standard deviation 0.25. The system output was corrupted by an additive zero-mean white Gaussian noise with standard deviation equal to 1. The system to identify is of the form $d_n = \psi_n(u_n)$.

The KAPA algorithm is used with the following parameters settings: number of Inputs/Outputs p = 3, step-size control parameter η = 0.01, regularization factor ϵ = 0.07 (see section 3 for details).The acronym AKAPA is adopted to indicate the dictionary Adaptation for Kernel Affine Projection Algorithm.

A set of 200 time series of 3000 samples each was used to compare the KAPA and AKAPA using the Normalized Mean Squared Error (NMSE) which was computed over the last 500 samples according to:

$$\text{NMSE} = E\left\{ \frac{\sum_{i=2501}^{3000} (d_n - \psi_n(u_n))^2}{\sum_{i=2501}^{3000} d_n^2} \right\}$$

Another indicator that must be computed is the average final size of the dictionary $\overline{m}$ calculated for the 200 time series. For the simulations, we adopted both the Gaussian kernel $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2/2\sigma^2)$ with a bandwidth $\sigma = 0.42$ and Exponential kernel $\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|/\sigma)$ with a bandwidth $\sigma = 0.33$.

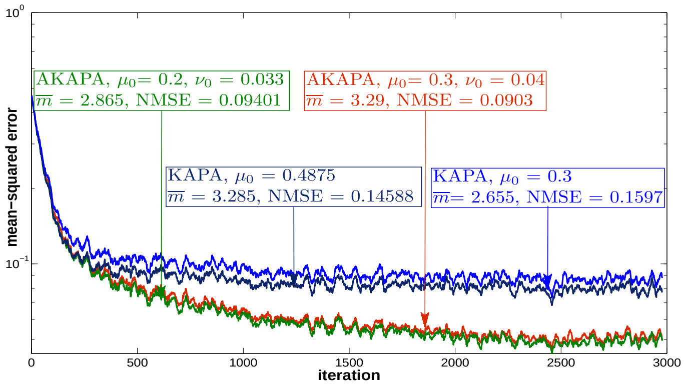

Fig. 2. Learning curves for KAPA and AKAPA using a Gaussian kernel (σ = 0.42) for different coherence parameters μ_0 and gradient step sizes ν_0.

The learning curves shown in Figure 2 and Figure 3 depict the Mean Squared Error and compare different settings for the coherence criterion μ_0 and for the adjustment step size ν_0. ν_0 has been selected using a rough grid search so as to obtain the best performances for the given value of μ_0. Table 1 gives a summary of the obtained results for the KAPA and the AKAPA algorithms. The obtained results lead to the following observations:

1. For the same coherence parameter $\mu_0 = 0.3$, AKAPA shows a 23.92% increase in the average size of the dictionary $\overline{m}$ with a decrease of 41.60% in the NMSE for the Gaussian kernel (12.12% and 16.93% respectively for the Exponential kernel). See rows 1 and 3 in Table 1.

[3] This property is due to the fact that the kernel function satisfies the validity condition (8), namely $f^{(1)}(\|\delta\boldsymbol{u}\|^2) \leq 0$, and $\mu_0 \geq f(\|\delta\boldsymbol{u}\|^2)$ as the coherence constraint is satisfied at any time step.

Table 1. Experimental Setup and Performance, with $p = 3$, $\eta = 0.01$, and $\epsilon = 0.07$

	Gaussian kernel ($\sigma = 0.42$)			Exponential kernel ($\sigma = 0.33$)		
Algorithm	Parameter Settings	$\overline{m}$	NMSE	Parameter Settings	$\overline{m}$	NMSE
KAPA [10]	μ_0=0.3	2.655	0.15970	μ_0=0.3	5.61	0.15392
KAPA [10]	μ_0=0.49	3.285	0.14588	μ_0=0.345	6.26	0.15070
AKAPA (this paper)	μ_0=0.3, ν_0=0.04	3.29	0.090327	μ_0=0.3, ν_0=0.01	6.29	0.12786
AKAPA (this paper)	μ_0=0.2, ν_0=0.033	2.865	0.09401	μ_0=0.25, ν_0=0.005	5.41	0.12751

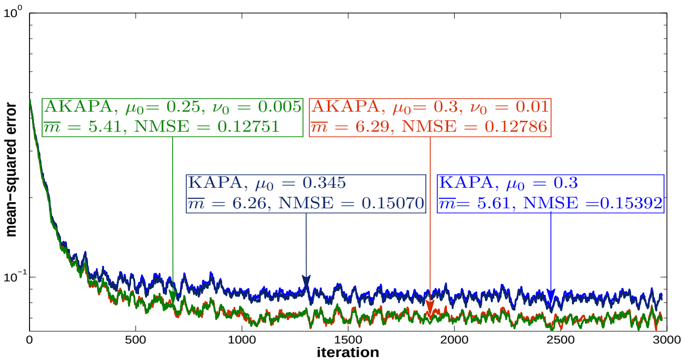

Fig. 3. Learning curves for KAPA and AKAPA using an Exponential kernel ($\sigma = 0.33$) for different coherence parameters μ_0 and gradient step sizes ν_0.

2. Comparing NMSE for the same average sizes of the dictionaries $\overline{m}$ using different values for μ_0, AKAPA led to a decrease of 38.08% for the Gaussian kernel and 15.15% for the Exponential kernel. See rows 2 and 3 in Table 1.

From these observations we can deduce that, if μ_0 and ν_0 were properly selected, the size of the dictionary and the approximation error can be greatly reduced. These results were also observed on other (synthetic and real) time series, omitted here due to space limitations.

6. CONCLUSION

In this paper, we demonstrated the interest of adjusting the dictionary elements within the context of online predictions with kernel-based methods. Our idea was to use an iterative gradient adaptation algorithm that satisfies a coherence measure for the elements of the dictionary. Pruning the dictionary to reduce its size, as well as other adaptation algorithms, will be considered in our future work.

7. REFERENCES

[1] S. Haykin, *Neural Networks and Learning Machines (3rd Edition)*, Prentice Hall, 3rd edition, Nov. 2008.

[2] M. Schetzen, *The Volterra and Wiener theories of nonlinear systems*, Wiley, 1980.

[3] N. Wiener, *Nonlinear Problems in Random Theory (Technology Press Research Monographs)*, The MIT Press, Aug. 1966.

[4] A. Smola and B. Schölkopf, "A tutorial on support vector regression," NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998, To appear in Statistics and Computing, 2001.

[5] J.A.K. Suykens, T. van Gestel, J. de Brabanter, B. deMoor, and J. Vandewalle, *Least squares support vector machines*, World Scientific, 2002.

[6] T. J. Dodd, V. Kadirkamanathan, and R. F. Harrison, "Function estimation in hilbert space using sequential projections," *Intell. Control Syst. Signal Process.*, pp. 113–118, 2003.

[7] S. Phonphitakchai and T.J. Dodd, "Stochastic meta descent in online kernel methods," in *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, may 2009, vol. 02, pp. 690 –693.

[8] Y. Engel, S. Mannor, and R. Meir, "Kernel recursive least squares," *IEEE trans. on Signal Processing*, vol. 52, pp. 2275–2285, 2004.

[9] P. Honeine, C. Richard, and J. C. M. Bermudez, "On-line nonlinear sparse approximation of functions," in *Proc. IEEE International Symposium on Information Theory*, Nice, France, June 2007, pp. 956–960.

[10] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE trans. on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, March 2009.

[11] G. S. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82–95, 1971.

[12] B. Schölkopf, R. Herbrich, and A. Smola, "A generalized representer theorem.," in *COLT/EuroCOLT'01*, 2001, pp. 416–426.

[13] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2231–2242, 2004.

[14] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electron. Commun. Japan*, vol. 67, no. A, pp. 1927–1984, 1984.

[15] A. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, New York, 2003.

[16] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, Wiley Publishing, 1st edition, 2010.

[17] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bulletin of the American Mathematical Society*, vol. 39, pp. 1–49, 2002.