

# APPLICATIONS OF COMPLEX AUGMENTED KERNELS TO WIND PROFILE PREDICTION

*Anthony Kuh*

University of Hawaii  
Department Electrical Engineering  
Honolulu, HI 96822  
kuh@hawaii.edu

*Danilo Mandic*

Imperial College London  
Department Electrical and Electronic Engineering  
Exhibition Rd., London SW72BT  
d.mandic@imperial.ac.uk

**Index Terms:** Complex augmented kernels, wind prediction

## ABSTRACT

This paper combines complex signal processing with kernel methods for applications in wind prediction. Specifically, we consider developing least squares kernel algorithms for both complex data and augmented complex data. The augmented complex kernel algorithms have advantages over complex kernel algorithms in both the areas of performance and complexity. Use of kernels also allow implementation of nonlinear algorithms by working in the dual space. We apply our algorithm to wind series time prediction and show that our augmented complex algorithms outperform other complex least square algorithms.

## 1. INTRODUCTION

In recent years there has been a renewed interest in complex signal processing. Research in [11, 12] give a rigorous discussion of proper and improper complex vectors and processes. In [5] there is a discussion of complex neural network algorithms and applications. Complex signal processing is now being applied to problems in communications [12], imaging, remote sensing, renewable energy [9], and biomedical processing [8]. This paper discusses implementation of complex and augmented complex algorithms for least squares kernel methods. We demonstrate the performance of these algorithms on wind data which is improper.

Predicting wind speed and direction is becoming increasingly more important as we improve development of this source of renewable energy. Accurate wind prediction can ensure the efficient operation of wind turbines and wind farms. Short term prediction is crucial to damage protection and vibration control of wind turbines. Medium and long term prediction can help with the integration of wind energy to the power grid [9]. The power wind turbines generates is difficult to forecast, because of fluctuations in wind speed and direction. The impact of wind direction becomes more prominent when wind is milder than in strong winds as placement of turbines becomes more critical [7]. Previous research has found that using the

augmented complex LMS algorithm can more accurately predict wind speed and energy [9]. We extend this work by applying complex kernel methods.

Kernel methods have become a popular tool to use in many applications, because they can be solved via convex optimization methods, nonlinear processing can be performed, and the solution is often found in the dual space via the kernel trick [2, 10, 14]. In signal processing applications and time series prediction it is often necessary to implement algorithms that are online and adaptive. Kernel methods can easily be implemented in an online recursive way when the algorithms use a least squares cost function [6, 13]. This is because the solution involves solving a set of linear equations that can be solved in primal feature space or the kernel observation space using recursive adaptive filtering methods [4]. The solution depends on some subset of the observation data which are called support vectors. A major problem with kernel least squares methods is that the solution is not sparse as all observation data are support vectors [13]. The solution can be made sparse in the number of support vectors by using various methods including constrained subspace approaches [3, 6, 1]. For these methods the weight vector is constrained to lie in the subspace generated by the support vectors.

Here we formulate a complex augmented kernel subspace least squares algorithm. Section 2 discusses implementing complex augmented kernel methods. We first look at the sample complex correlation matrix and the sample augmented correlation matrix. We then examine the complex kernel matrix and the complex augmented kernel matrix. In Section 3 we discuss the complex augmented subspace kernel regression algorithm. Section 4 discusses issues associated with online implementation of the complex augmented subspace kernel regression algorithm. In Section 5 we look at applying online algorithms to wind prediction. In conducting simulations we consider performance, at different sampling rates for wind classified that have 'high', 'medium', and 'low' speeds. Section 6 summarizes results of this paper.

## 2. COMPLEX AUGMENTED KERNELS

Let  $z(i) = z_R(i) + jz_I(i) \in \mathcal{C}^d$  with  $1 \leq i \leq m$ . The data can be rewritten compactly as  $\mathbf{Z}$  where  $\mathbf{Z} = [z(1)|\dots|z(m)]$ .  $\mathbf{Z}$  is known as the data matrix and can be expressed in terms of real and imaginary parts as  $\mathbf{Z} = \mathbf{Z}_R + j\mathbf{Z}_I$ . For this data, the complex correlation matrix is defined as

$$R = \mathbf{Z}\mathbf{Z}^H = \mathbf{Z}_R\mathbf{Z}_R^T + \mathbf{Z}_I\mathbf{Z}_I^T + j(\mathbf{Z}_I\mathbf{Z}_R^T - \mathbf{Z}_R\mathbf{Z}_I^T). \quad (1)$$

We can also define the complex kernel matrix as

$$K = \mathbf{Z}^H\mathbf{Z} = \mathbf{Z}_R^T\mathbf{Z}_R + \mathbf{Z}_I^T\mathbf{Z}_I + j(\mathbf{Z}_R^T\mathbf{Z}_I - \mathbf{Z}_I^T\mathbf{Z}_R) \quad (2)$$

In cases where the complex process is not proper [12] we can augment the data matrix to create the augmented data matrix

$$\mathbf{Z}_a = \begin{bmatrix} \mathbf{Z} \\ \mathbf{Z}^* \end{bmatrix}. \quad (3)$$

We can then get the augmented complex correlation matrix given by

$$R_a = \mathbf{Z}_a\mathbf{Z}_a^H = \begin{bmatrix} R & C \\ C^* & R^* \end{bmatrix} \quad (4)$$

where

$$C = \mathbf{Z}\mathbf{Z}^T = \mathbf{Z}_R\mathbf{Z}_R^T - \mathbf{Z}_I\mathbf{Z}_I^H + j(\mathbf{Z}_I\mathbf{Z}_R^T + \mathbf{Z}_R\mathbf{Z}_I^T). \quad (5)$$

and the augmented kernel is given by

$$K_a = \mathbf{Z}_a^H\mathbf{Z}_a = K + K^* = 2(\mathbf{Z}_R^T\mathbf{Z}_R + \mathbf{Z}_I^T\mathbf{Z}_I) \quad (6)$$

which is a real kernel matrix. When performing least squares regression with  $z$  describing the observation vector, we can work with either the kernel matrix or the correlation matrix. For improper complex data more information is gotten from the augmented case where we can work with the augmented kernel matrix  $K_a$  or the augmented correlation matrix  $R_a$ .

When working in primal space with correlation matrices it may be more advantageous to work with the complex correlation matrix  $R$  rather than the augmented complex correlation matrix  $R_a$ . This is especially true if data is proper or circular (i.e.  $E(C) = 0$ ) as more processing will be required with  $R_a$  as it is  $2d \times 2d$  versus  $R$  which is  $d \times d$ .

In many cases there are distinct advantages with working with kernel matrices as opposed to correlation matrices as we are interested in more flexibility, the input dimension is high, or we want to work with nonlinear kernels. When working with kernels it is always advantageous to work with the complex augmented kernel matrix rather than the complex kernel matrix. For improper data performance will be better, both the complex and augmented complex kernel matrix have the same dimension, and computations are less for the complex augmented kernel matrix as all entries are real whereas they are complex for the complex kernel matrix (i.e. see equations (2) and (6)). From here on, when we discuss kernel methods we will work with the augmented complex case.

## 3. AUGMENTED LEAST SQUARES SUBSPACE REGRESSION

In this section we develop an augmented complex least-squares subspace kernel algorithm using the least squares Support Vector Machine developed in [13] and the subspace method presented in [6]. Here we are given training examples  $(x(i), y(i))$ ,  $1 \leq i \leq m$  where  $x(i) \in \mathcal{C}^n$  and  $y(i) \in \mathcal{C}$ . We represent the data compactly as  $(\mathbf{x}, \mathbf{y})$  where  $\mathbf{x} = [x(1)|\dots|x(m)]$  and  $\mathbf{y} = [y(1), \dots, y(m)]^T$ . The inputs are transformed from input space to feature space via kernel functions  $\phi(x)$  that map inputs from  $\mathcal{C}^n$  to feature space  $\mathcal{C}^d$ . Let  $\mathbf{Z} = \Phi(\mathbf{x}) = [\phi(x(1))|\dots|\phi(x(m))]$ . Then using the augmented system we can formulate a least squares optimization problem given by

$$\min J(w_a, b) = \min_{w_a, b} \frac{1}{2}\|w_a\|^2 + \frac{\gamma}{2}\|\mathbf{e}\|^2 \quad (7)$$

subject to

$$\mathbf{e} = \mathbf{y} - \mathbf{Z}_a^H w_a - \mathbf{1}b \quad (8)$$

where  $w_a \in \mathcal{C}^{2d}$  is the augmented weight vector,  $\mathbf{1}$  is a vector of 1s, and  $b \in \mathcal{C}$  is the complex threshold value. Equation (7) contains two terms with the first term controlling complexity and the second term controlling squared error with  $\gamma$  denoting the regularization value that weights the squared error. The subspace method picks a subset of  $m_s$  columns of  $\mathbf{Z}_a$  to form a matrix  $\mathbf{Z}_s$ . The training inputs associated with these columns are the support vectors and methods for choosing these support vectors are discussed in [6]. The augmented weights are constrained to lie in the subspace generated by the chosen columns and this constraint can be expressed as

$$w_a = \mathbf{Z}_s \alpha. \quad (9)$$

where  $\alpha$  is a complex-valued  $m_s$  vector weighting the training feature vectors. Define  $K_{SS} = \mathbf{Z}_s^H \mathbf{Z}_s$  and  $K_S = \mathbf{Z}_s^H \mathbf{Z}_a$  and note that these matrices like the matrix  $K_a$  are all real for the augmented case. By substituting equations (8,9) in equation (7) we have that

$$\min Q(\alpha, b) = \min \frac{1}{2}\alpha^H K_{SS}\alpha + \frac{\gamma}{2}\|\mathbf{y} - K_S^H \alpha - \mathbf{1}b\|^2. \quad (10)$$

This problem is solved by finding the solution to the following set of linear equations,

$$\begin{bmatrix} m & \mathbf{1}^T K_S^H \\ K_S \mathbf{1} & K_{SS}/\gamma + K_S K_S^H \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{1}^T \mathbf{y} \\ K_S \mathbf{y} \end{bmatrix}. \quad (11)$$

Assume  $A = K_{SS}/\gamma + K_S K_S^H$  is invertible. By elimination we then get that

$$b = \frac{\mathbf{1}^T \mathbf{y} - \mathbf{1}^T K_S^H A^{-1} K_S \mathbf{y}}{m - \mathbf{1}^T K_S^H A^{-1} K_S \mathbf{1}} \quad (12)$$

and

$$\alpha = A^{-1} K_S (\mathbf{y} - \mathbf{1}b). \quad (13)$$

To get the estimated output for a given input  $x$ , let  $z = \Phi(x)$  and

$$z_a = \begin{bmatrix} z \\ z^* \end{bmatrix}.$$

The estimated output is then given by

$$\hat{y}(z) = w_a^H z_a + b. \quad (14)$$

Substituting equation (9) into the above equation we then get that

$$\hat{y}(z) = \alpha^H K_{Sz} + b \quad (15)$$

where  $K_{Sz} = \mathbf{Z}_s^H z_a$ .

#### 4. ONLINE AUGMENTED KERNEL ALGORITHMS

We consider prediction of time series data. Data is given by a sequence of complex data,  $x(n) \in \mathcal{C}$ . Here the input  $z(n) = [x(n) \dots x(n-d+1)]$  and the associated output is  $y(n) = x(n+L)$  where  $L \geq 1$  is the prediction step. For kernel algorithms we consider online recursive algorithms where at time  $n$  the support vector parameters are given by  $\alpha(n)$  and the threshold value is given by  $b(n)$ . The estimate of  $y(n)$  is

$$\hat{y}(n) = \alpha(n)^H K_{Sz(n)} + b(n) \quad (16)$$

The kernel online algorithm is then given the output  $y(n)$  and updates the data matrices to compute the inverse of  $A$  and then update the support vector and threshold value. In addition to the length of the input vector,  $d$  we must determine the number of support vectors  $m_s$  and the number of information vectors  $m$  that are used in the kernel subspace algorithm.

For the kernel subspace algorithm we need a method to choose the  $m_s$  support vectors. For time series prediction it is reasonable to choose the  $m_s$  most recent observation data as support vectors. Similarly the  $m > m_s$  information vectors are chosen from the most recent observations. The online kernel algorithms require more storage space than recursive least squares algorithms as we need to store the  $m$  information vectors in addition to data information matrices. This is balanced against the advantages of kernel methods; ability to perform nonlinear processing, more flexibility in the design of kernel methods, and the advantages with working in the dual observation space.

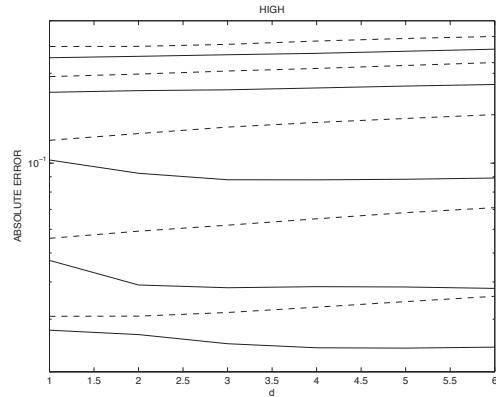
#### 5. WIND PREDICTION USING COMPLEX ALGORITHMS

We examined wind data sampled at 50Hz in an urban environment [9]. The data was divided into three data sets; 'high', 'medium', and 'low'. Data was sampled over a one day period. For this time series data we formed an input vector and considered one step prediction. Data contained wind in East and North directions which were mapped on to a complex vector. We considered recursive online algorithms where at

each time instant the algorithm first made a prediction, was given the correct output, and the new data vector.

Simulations were conducted for a complex RLS (CRLS) algorithm as performance was similar to the augmented complex RLS algorithm with less computations. For kernel algorithms we used the augmented complex subspace kernel (ACSK) algorithm as it gave better performance than the complex subspace kernel algorithm with less computations. Here we show simulations where data was first averaged over ten data samples. We then downsampled this averaged data at rates of every (1,2,5,10, and 15) samples. Sampling at lower rates is important to save on communication and complexity costs. Here the prediction step  $L$  is the sampling rate.

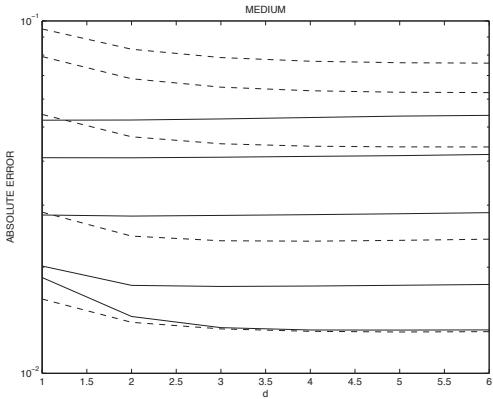
Results are shown in fig. 1 for 'high' data, fig. 2 for 'medium' data, and fig. 3 for 'low' data. When all data was taken there was not much difference between ACSK and CRLS performance. ACSK performance was slightly better than CRLS performance for 'high' data. When data was sampled at intervals of (2,5,10, and 15) data points ACSK performed better than CRLS. ACSK stores input data more effectively than CRLS. The degradation from sampling less frequently in ACSK is less than that from CRLS. For 'high' data, absolute error rates are higher and low order filters perform about the same as high order filters. Here the performance gaps between sampled data at intervals of (2,5,10 and 15) data points is less. For 'low' data there was larger gaps in performance between ACSK and CRLS for sampled data at intervals of (2,5,10 and 15) data points. This is because more information could be captured about relationships between different data using kernels.



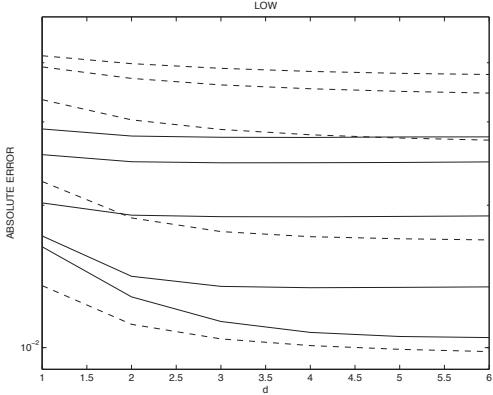
**Fig. 1.** 'High' data. Average of ten data. Solid lines (dashed lines) are ACSK (CRLS) with curves from bottom to top representing sampling every (1,2,5,10,15) points.

#### 6. SUMMARY

This paper implements an augmented complex subspace kernel algorithm using a least squares cost criteria. This algo-



**Fig. 2.** 'Medium' data. Average of ten data. Solid lines (dashed lines) are ACSK (CRLS) with curves from bottom to top representing sampling every (1,2,5,10,15) points.



**Fig. 3.** 'Low' data. Average of ten data. Solid lines (dashed lines) are ACSK (CRLS) with curves from bottom to top representing sampling every (1,2,5,10,15) points.

rithm has a number of desirable properties in that the augmented kernel matrix is real, performance is superior to a complex subspace kernel algorithms, nonlinear kernels can easily be implemented, the algorithm can be implemented in an online recursive manner, and the algorithm provides a lot of flexibility in its design.

We apply the complex algorithms to the prediction of wind. The augmented kernel subspace algorithm is shown to have similar or better performance than recursive least squares (RLS) algorithms working with correlation or augmented correlation matrices. The kernel algorithms require more storage than RLS algorithms, but have more flexibility and performance is better when data is averaged and sampled. This is true for 'high', 'medium', and 'low' wind data.

## 7. REFERENCES

- [1] G. C. Cawley, N. L. C. Talbot. A greedy training algorithm for sparse least-squares support vector machines, *Proceedings of the International Conference on Artificial Neural Networks ICANN 2002*, Madrid, Spain, 681–686, 2002.
- [2] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, U.K., 2000.
- [3] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm, *IEEE Trans. Signal Proc.*, Vol. 52, 8, 2275-2285, Aug., 2004.
- [4] S. Haykin. *Adaptive filter theory*, 4th Ed., Prentice Hall, Englewood Cliffs, NJ, 2003.
- [5] *Complex valued neural networks: theory and applications*, Ed. A. Hirose, World Scientific, 2003.
- [6] A. Kuh, "Intelligent recursive kernel subspace estimation algorithms," in *Proceedings of the 39th annual Conference on Information Sciences and Systems (CISS 2005)*, 216-221, Baltimore, MD, USA, Mar., 2005.
- [7] S. Li, D. Wunsch, E. OHair, M. Giesselmann . "Using NNs to estimate wind turbine power generation," *IEEE Transaction on Energy Conversion* 2001;16(3): 27682
- [8] *Signal Processing Techniques for Knowledge Extraction and Information Fusion*, D. Mandic, M. Golz, A. Kuh, D. Obradovic, and T. Tanaka, Editors, Springer, to be published, Oct., 2007.
- [9] D. P. Mandic, S. Javidi, S. L. Goh, A. Kuh and K. Aihara, "Complex Valued Prediction of Wind Profile Using Augmented Complex Statistics", *Renewable Energy*, Vol 34, 1, 196-201, 2009.
- [10] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An Introduction to Kernel-Based Learning Algorithms. *IEEE Trans. on Neural Networks*, Vol. 12, #2, 181-202, 2001.
- [11] F. Neeser, J. Massey, "Proper complex random processes with applications to information theory", *IEEE Trans. Inform. Theory*, vol. 39, 1293-1302, July, 1993.
- [12] P. Schreier and L. Scharf. Second-Order Analysis of Improper Complex Random Vectors and Processes, *IEEE Trans. on Signal Processing* Vol. 51, 3, 714-725, Mar. 2003.
- [13] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least squares support vector machines*, World Scientific Publishing Co., Singapore, 2000.
- [14] V. Vapnik. *Statistical learning theory*. John Wiley, New York City, NY, 1998.