URSI GASS 2023, Sapporo, Japan, 19 – 26 August 2023

Efficient Implementation of the Vector-Valued Kernel Ridge Regression for the Parametric Modeling of the Frequency-Response of a High-Speed Link

N. Soleimani[(1)] and R. Trinchero[(1)]
(1) Politecnico di Torino, Torino, Italy; e-mail: nastaran.soleimani@polito.it; riccardo.trinchero@polito.it

Abstract

This paper presents an efficient implementation of the vector-output kernel Ridge regression (KRR). The proposed approach is applied to model the frequency-domain behavior of the magnitude of the transfer function of a parametric high-speed link as function of 11 parameters. The accuracy and the computational cost of the proposed technique are assessed on noisy samples and compared with the ones of a state-of-the-art modeling technique based on the combination of the principal components analysis (PCA) and the least-squares support vector machine (LS-SVM) regression.

1 Introduction

During the last decades, machine learning (ML) techniques have emerged as one of the top approaches for regression and classification problems with a widely range of applications. Within the electrical and electronic field, kernel-based ML regressions, such as (SVM) regression [1], least-squares support vector machine (LS-SVM) regression [2], kernel Ridge regression (KRR) and its variants [3, 4], have shown interesting performance for the modeling of electronic and electromagnetic (EM) structures. The above techniques allow to build accurate and fast-to-evaluate parametric models, also known as surrogate models, of the responses of electronic devices and EM structures starting from a small set of training samples. The obtained surrogate models are known in closed-form, thus providing a fast-to-evaluate and efficient alternative to computer experiments (e.g., simulations) in computationally expensive design tasks such as uncertainty quantification and optimization [5].

Indeed, kernel-machine regressions allow to build non-parametric surrogate models, in which the number of unknowns are independent from the number of input parameters considered by the model [5]. Moreover, they rely on a linear model structure in which the model unknowns can be estimated from the solution of a convex optimization problem [6], thus leading to a faster training time and improved accuracy with respect to regression model trained via artificial neural network (ANN) [2, 7].

On the other hand, different from ANN structure, standard formulation of kernel machine regressions is limited to single-output problem, thus making their direct application to multi-output scenarios rather cumbersome. Unfortunately, multi-output or vector-valued problems are quite common in electronic applications. For instance, we might be interested to model the parametric frequency- or time-domain behavior of an electronic device as a function of its internal parameters. The above problem can be tackled via a scalar-valued regression, but it would require to train a possible *huge* number of uncorrelated scalar-output models, one for each output components (i.e., frequency or time samples). Moreover, the above approach unavoidably ignores any correlation among the output components, compromising its accuracy and robustness to noise [8]. A clever workaround to the above issues consists in compressing the output-dimension via a compression techniques such as the principal component analysis (PCA). The resulting compressed representation of the output components allows to heavily reduce the number of single output regression problems to be solved, with beneficial effects on the training cost [9]. However, such manipulation of the training set can lead to generalization issues with a possible leak of accuracy, when a small set of components is considered [3].

As an alternative to the above approach, a generalized multi-output formulation of the KRR has been presented in [3, 4]. Such approach can be directly applied to tackle multi-output regression problems without requiring any data manipulation. However, despite its improved accuracy with respect to state-of-the-art approaches based on data compression, a plain implementation of the such generalized KRR had shown a high training cost. Indeed, for the former methods the model training requires the solution of a large linear system.

This paper presents an efficient implementation of the vector-valued KRR (inspired by [10]) based on the diagonalization of its constitutive equations. The effectiveness and the robustness to noise of a proposed technique are investigated on the prediction of the magnitude of the frequency response of a high-speed link with 11 uniform distributed parameters by considering a noisy training set. The model performance are then compared with the ones of a state-of-the-art technique combining PCA compression with the LS-SVM regression [9].

2 Vector-Valued Kernel Ridge Regression

Let us consider the problem of building a generic vector-valued surrogate model $\hat{\boldsymbol{f}} : \mathcal{X} \to \mathcal{Y}$, starting from the information available on the training set $\mathcal{D} = \{(\boldsymbol{x}_l, \boldsymbol{y}_l\}_{l=1}^{L}$, such that $\boldsymbol{x}_l \in \mathcal{X} \subseteq \mathbb{R}^p$ and $\boldsymbol{y}_l \in \mathcal{Y} \subseteq \mathbb{R}^D$. The above problem turns out to be equivalent to learn D scalar functions $\hat{f}^{(d)} : \mathcal{X} \to \mathbb{R}$ with $d = 1, \ldots, D$ minimizing the following empirical risk functional:

$$\hat{\boldsymbol{f}} = \underset{\tilde{\boldsymbol{f}} \in \mathcal{H}}{\arg\min} \sum_{d=1}^{D} \sum_{l=1}^{L} (y_l^{(d)} - \tilde{f}^{(d)}(\boldsymbol{x}_l))^2 + \lambda \|\tilde{\boldsymbol{f}}\|_{\mathcal{H}}^2, \quad (1)$$

where λ is the regularizer hyperparameter providing a trade-off between the model flatness and accuracy on the training set, whilst $y_l^{(d)}$ and $\tilde{f}^{(d)}(\boldsymbol{x}_l)$ represent the d-th component of the $l-$th training output and the corresponding model prediction, respectively.

According to the represented theorem for vector-valued regression problem presented in [11], any optimal solution $\hat{\boldsymbol{f}}$ of (1) writes:

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{l=1}^{L} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l) \boldsymbol{c}_l, \quad (2)$$

where $\hat{\boldsymbol{f}}(\boldsymbol{x}) = [\hat{f}^{(1)}(\boldsymbol{x}), \ldots, \hat{f}^{(D)}(\boldsymbol{x})]^T$ is a vector collecting the model prediction for any $\boldsymbol{x} \in \mathcal{X}$, $\mathbf{K}(\cdot,\cdot) : \mathbb{R}^{p \times p} \to \mathbb{R}^{D \times D}$ is a multi-output kernel matrix and $\boldsymbol{c}_l = [c_{1,l}, \ldots, c_{D,l}]^T \in \mathbb{R}^D$ are column vectors collecting the regression unknowns.

Hereafter in this paper, we will consider the following separable structure for the matrix kernel function $\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')$:

$$[\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')]_{[d,d']} = k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}') k_o(d, d'), \quad (3)$$

where $k_{\boldsymbol{x}}$ and k_o are scalar kernels acting independently on the input space (i.e., $k_{\boldsymbol{x}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$) and on the output dimensions (i.e., $k_o : \{1, \ldots, D\} \times \{1, \ldots, D\} \to \mathbb{R}$).

By using (2) and (3), the empirical risk minimization in (1) can be recast in terms of the following discrete-time Sylvester equation:

$$\mathbf{K}_{\boldsymbol{x}} \mathbf{C} \mathbf{B} + \lambda \mathbf{C} = \mathbf{Y}, \quad (4)$$

where $\mathbf{K}_{\boldsymbol{x}}$ is a $L \times L$ Gram matrix computed from the input samples $\{\boldsymbol{x}_l\}_{l=1}^{L}$ (i.e., $[\mathbf{K}_{\boldsymbol{x}}]_{ij} = k_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$), $\mathbf{B}$ is a $D \times D$ Gram matrix computed on the output dimensions $\{1, \ldots, D\}$ (i.e.,$[\mathbf{B}]_{ij} = k_o(d_i, d_j)$), $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L]^T \in \mathbb{R}^{L \times D}$ is a matrix collecting the model unknowns and $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l]^T$ is a $L \times D$ matrix associated to the training output.

By using the properties of the Kronecker product, the solution of the discrete-time Sylvester equation in (4) can be recast as the solution of a linear system of equations [12]. Unfortunately, the above formulation leads to a *huge* linear system with $(LD) \times (LD)$ equations, which solution would require a computation cost proportional to $\mathcal{O}(L^3 D^3)$. Such training cost can be heavily reduced by diagonalizing the kernel matrices $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$ [10], i.e.,:

$$\mathbf{K}_{\boldsymbol{x}} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T \text{ and } \mathbf{B} = \mathbf{T} \mathbf{M} \mathbf{T}^T, \quad (5)$$

where $\mathbf{U} \in \mathbb{R}^{L \times L}$ and $\mathbf{T} \in \mathbb{R}^{D \times D}$ are matrices collecting the eigenvectors of the matrices $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$, respectively, whereas $\boldsymbol{\Lambda} \in \mathbb{R}^{L \times L}$ and $\mathbf{M} \in \mathbb{R}^{D \times D}$ are diagonal matrices collecting the corresponding eigenvalues. Using the definitions in (5), the Sylvester equation in (4) can be rewritten as:

$$\boldsymbol{\Lambda} \tilde{\mathbf{C}} \mathbf{M} + \lambda \tilde{\mathbf{C}} = \tilde{\mathbf{Y}} \quad (6)$$

where $\tilde{\mathbf{C}} = \mathbf{U}^T \mathbf{C} \mathbf{T}$ and $\tilde{\mathbf{Y}} = \mathbf{U}^T \mathbf{Y} \mathbf{T}$ are new transformed matrices collecting a transformed version of regression unknowns and source term, respectively. Due to the diagonal structure of (6), a generic entry of the unknown matrix $[\tilde{\mathbf{C}}]_{ij} = c_{ij}$ can be suitably computed via a scalar equation defined by the diagonal eigenvector matrices $\boldsymbol{\Lambda}$ and $\mathbf{M}$, such as:

$$\tilde{c}_{ij} = \frac{\tilde{y}_{ij}}{[\boldsymbol{\Lambda}]_{ii} [\mathbf{M}]_{jj} + \lambda}. \quad (7)$$

Once the entries of the matrix $\tilde{\mathbf{C}}$ has been computed via the above equation, the original unknown matrix $\mathbf{C}$ can be reconstructed as:

$$\mathbf{C} = \mathbf{U} \tilde{\mathbf{C}} \mathbf{T}^T. \quad (8)$$

The above approach for solving the discrete-time Sylvester equation turns out to be more efficient than the equivalent solution based on the Kronecker formulation [12] presented in [3]. Indeed, since the diagonalization is applied on the matrices $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$ separately, the computational cost required for the model training reduces from $\mathcal{O}(L^3 D^3)$ to $\mathcal{O}(L^3 + D^3 + L^2 D + L D^2)$, thus leading to beneficial effect on the training time when the product $L \times D$ is large.

3 Application Example: High-Speed link

The effectiveness of the proposed implementation of the vector-valued KRR are investigated on the prediction of the magnitude of the frequency response $H(\boldsymbol{x}; f) =$

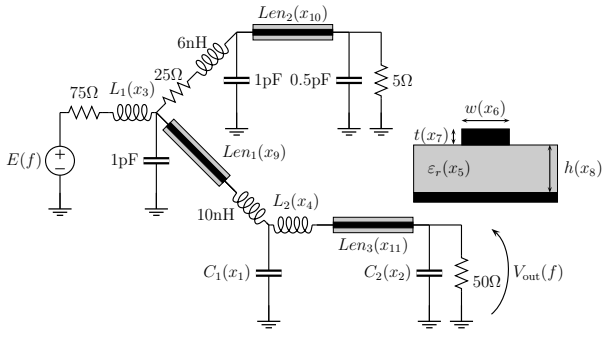

Figure 1. Structure of High-speed link [2] considered in Sec. 3.

Table 1. Mean value and corresponding relative range of variation of the 11 parameters of the high-speed in Fig. 1.

Parameter	Mean Value	Uniform Variation
$C_1(x_1)$	1 pF	50%
$C_2(x_2)$	0.5 pF	50%
$L_1(x_3)$	10 nH	50%
$L_2(x_4)$	10 nH	50%
$\varepsilon_r(x_5)$	4.1	1%
$w(x_6)$	252 μm	1%
$t(x_7)$	35 μm	1%
$h(x_8)$	60 μm	1%
$Len_1(x_9)$	5 cm	10%
$Len_2(x_{10})$	3 cm	10%
$Len_3(x_{11})$	3 cm	10%

$V_{out}(\boldsymbol{x};f)/E(f)$ of the high-speed link in Fig. 1, in a frequency bandwidth from 1 MHz to 2 GHz as a function of 11 parameters. The mean value and range of variability of the considered parameters are provided in Tab. 1.

Such variability is induced by means of the normalized parameters collected in the vector $\boldsymbol{x} = [x_1, \ldots, x_{11}]^T$, in which each parameter $x_i \sim \mathcal{U}([-1,+1])$ is considered as an uniformly distributed random variable. The accuracy and training time of the proposed vector-valued KRR are then compared with a state-of-the-art modeling technique consisting on the combination of the PCA and the LS-SVM regression [9].

The considered high-speed link structure has been implemented in MATLAB. The MATLAB implementation is used to generate the training and test pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, in which $\boldsymbol{y}_i = [|H(f_1;\boldsymbol{x}_i)|, \ldots, |H(f_D;\boldsymbol{x}_i)|]^T$, based on a latin hypercube sampling (LHS) scheme by considering $D = 150$ linearly spaced frequency points $\{f_k\}_{k=1}^{D}$.To stress the performance of the considered modeling approaches, the training output set $\{\boldsymbol{y}_l\}_{l=1}^{L}$ has been synthetically corrupted by an additive noise, i.e.,

$$y(f_j;\boldsymbol{x}_l) = |\Re\{H(f_j;\boldsymbol{x}_l)\}(1+\zeta_j^{\Re}) + \\ +\Im\{H(f_j;\boldsymbol{x}_l)\}(1+\zeta_j^{\Im})|, \qquad (9)$$

where $j = 1, \ldots, D$ and $\zeta_j^{\Im}, \zeta_j^{\Re} \sim \mathcal{U}([-\sigma_n, \sigma_n])$ represent a set of uncorrelated uniform distributed random variables defining the additive noise with a noise level $\sigma_n = 0.05$ affecting the real and imaginary part of the frequency response $H(\boldsymbol{x};f)$.

The obtained training set is then used to train two different surrogate models built via the proposed implementation of the vector-valued KRR and the PCA+LS-SVM regression. Similar to [3, 4], the the vector-valued KRR is trained by using a radial basis kernel for both input parameters and output dimensions. The model hyperparameters are tuned via a 3-fold cross-validation. For the model based on the PCA-LS-SVM regression two different compression levels have been considered by using a the relative tolerance of 0.6% and 0.01% for the PCA compression, leading to a compressed model with either 2 or 100 components, respectively.

Table 2. Relative L2-error and training time computed from the predictions in dB obtained by the proposed vector-valued KRR and PCA+LS-SVM regression trained with increasing number of noisy training samples.

Methods	$L = 30$		$L = 90$		$L = 150$	
	ε_{L2}	t_{train}	ε_{L2}	t_{train}	ε_{L2}	t_{train}
KRR (Proposed)	4.0%	25s	2.8%	38s	2.5%	51s
PCA+LS-SVM (Rel.Tol.=0.01%)	7.5%	6s	5.0%	33s	4.3%	63s
PCA+LS-SVM (Rel.Tol.=0.6%)	6.8%	1.5s	5.9%	1.5s	4.6%	2s

Table 2 provides an exhaustive comparison between the above methods in terms of training time and relative $L2$-norm error computed in dB on 1000 test samples for an increasing number of the training samples (i.e., $L = 30$, 90 and 150). The results show that the computational cost required to built a vector-valued model with the proposed efficient implementation of the multi-output KRR turns out to be comparable with the one required by the PCA+LS-SVM with a relative tolerance of 0.01%. It is important to notice that the proposed implementation of the vector-valued KRR has a speed up of $\times 30$ with respect to its plain implementation proposed in [4].

Concerning the model accuracy, the errors reported in the table clearly highlight the improved performance of the proposed vector-valued KRR with respect to the ones achieved by the PCA+LS-SVM for all the considered training sets. The above statement is further supported by the parametric plot in Fig. 2 computed from the predictions of the considered methods trained with $L = 150$ training samples for two random configurations of the input parameters belonging to the test set. The plot clearly highlights the limited capability of the PCA compression to learn the actual correlation among the output components when the data are corrupted by noise. Indeed, the compressed representation of the training set obtained from the PCA still contains a non-negligible level of noise, which cannot be filtered out even if a small number of components is considered. On the other hand, thanks to the output dimension regularization provided by the kernel k_o in (3), the corresponding model trained via the proposed vector-valued KRR turns out to be more accurate and robust to noise.

Moreover, Figure 3 shows a statistical comparison among the methods in terms of the probability density functions (PDFs) computed on 1000 test samples for all the frequency points. Also in this case, it is possible to notice the detrimental effect of the noise on the predictions obtained by the

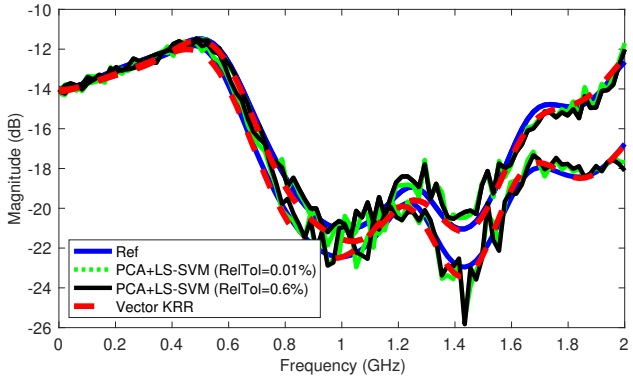

Figure 2. Parametric plots comparing the frequency responses predicted by the proposed method and the PCA+LS-SVM surrogate models for 2 different realizations of the input parameters.

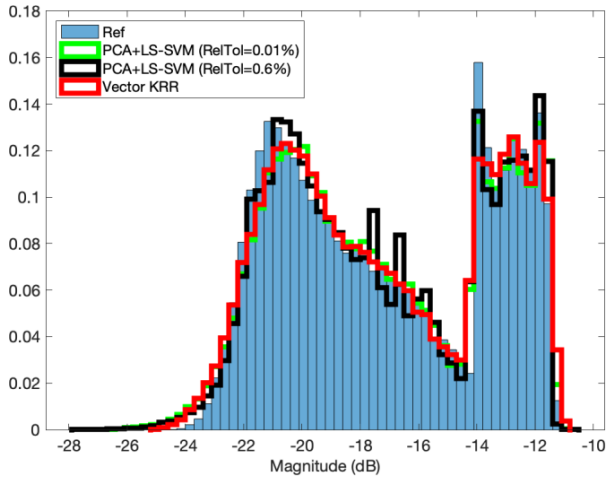

Figure 3. Comparison of the PDFs computed from the predictions of the surrogate models built via the proposed and PCA+LS-SVM regression with Tol.= 0.6% and 0.01% on 1000 test samples and for all the frequency points.

PCA+LS-SVM models, which is responsible for the spurious peaks visible in the corresponding PDFs around -18 and -16 dB.

4 Conclusions

This paper presented an efficient implementation of the vector-valued KRR.Its feasibility and performance have been investigated on a parametric and stochastic scenario by considering the magnitude of a frequency response of an high-speed link as a function of 11 parameters. The performance in terms of accuracy, computational cost and robustness to noise of the proposed technique are compared with the ones of a state-of-the-art modeling techniques based on the combination of the PCA and LS-SVM regression for noisy training samples. The results highlight the potentiality of the proposed method as well as its improved accuracy in noisy scenarios.

References

[1] R. Trinchero and F. G. Canavero, "Modeling of eye diagram height in high-speed links via support vector machine," in Proc. of 2018 IEEE 22nd Workshop on Signal and Power Integrity (SPI), Brest, 2018, pp. 1–4.

[2] N. Soleimani and R. Trinchero, "Compressed complex-valued least squares support vector machine regression for modeling of the frequency-domain responses of electromagnetic structures," *Electronics*, vol. 11, no. 4, 2022.

[3] N. Soleimani, R. Trinchero and F. Canavero, "Vector-Valued Kernel Ridge Regression for the Modeling of High-Speed Links," in Proc. IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO 2022), Limoges (France), July 6-8.

[4] N. Soleimani, R. Trinchero and F. Canavero, "Bridging the Gap Between Artificial Neural Networks and Kernel Regressions for Vector-Valued Problems in Microwave Applications," *IEEE Transactions on Microwave Theory and Techniques* (submitted).

[5] R. Trinchero and F. Canavero, "Machine Learning Regression Techniques for the Modeling of Complex Systems: An Overview," *IEEE Electromagnetic Compatibility Magazine*, vol. 10, no. 4, pp. 71-79, 4th Quarter 2021

[6] A. Rudi, L. Carratino, and L. Rosasco, "Falkon: An optimal large scale kernel method," *Advances in neural information processing systems*, 30, 2017.

[7] S. Kushwaha, et. al,"Comparative Analysis of Prior Knowledge-Based Machine Learning Metamodels for Modeling Hybrid Copper–Graphene On Chip Interconnects," *IEEE Transactions on Electromagnetic Compatibility*, vol. 64, no. 6, pp. 2249-2260, Dec. 2022.

[8] A. Mauricio, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends in Machine Learning* 4(3) pp. 195-266, 2012.

[9] P. Manfredi and R. Trinchero, "A data compression strategy for the efficient uncertainty quantification of time-domain circuit responses," *IEEE Access*, vol. 8, pp. 92019–92027, 2020.

[10] L. Baldassarre, et al., "Multi-output learning via spectral filtering", *Machine Learning*, vol. 87, pp. 259–301, 2012.

[11] C.A. Micchelli and M. Pontil, "On learning vector–valued functions," *Neural Computation*, vol. 17, pag. 177-–204, 2005.

[12] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge: Cambridge University Press, 1991.