Vector-Valued Kernel Ridge Regression for the Modeling of High-Speed Links

Vector-Valued Kernel Ridge Regression for the Modeling of High-Speed Links

Nastaran Soleimani, Riccardo Trinchero and Flavio Canavero
EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
E-mail: {nastaran.soleimani,riccardo.trinchero,flavio.canavero}@polito.it

***Abstract*—This paper presents a preliminary implementation of a general modeling framework for vector-valued functions based on a multi-output kernel Ridge regression (KRR). The proposed approach is based on a generalized definition of the reproducing kernel Hilbert space (RKHS) to the case of vector-valued functions, thus bridging the gap between multi-output Neural Network (NN) structures and standard scalar kernel-based approaches. The above concept is then used within the KRR to train a multi-output surrogate model able to predict the frequency responses of a high-speed link affected by four parameters with a large variability. The performance of the proposed approach, in terms of parametric and stochastic analysis, is compared with the one provided by two state-of-the-art techniques, such as the combination of the principal components analysis (PCA) and the least-squares support vector machine (LS-SVM) regression and a multi-output feed-forward NN structure.**

***Index Terms*—Kernel, reproducing kernel Hilbert space, Kernel Ridge Regression, high-speed link, parametric modeling, stochastic analysis**

I. Introduction

In the last decades, Machine Learning (ML) kernel-based regressions, such as support vector machine (SVM) regression [1], [2], least-squares support vector machine (LS-SVM) regression [2], [3], and kernel Ridge Regression (KRR) and its variants [4], have been successful adopted to construct accurate and fast to evaluate parametric models of the responses of complex electromagnetic (EM) structures. The resulting model, usually referred as "surrogate model", can be suitably adopted within uncertainty quantification and optimization task, thus proving an efficient alternative to the plain space exploration approaches based on a set of parametric simulations via the expensive full-computation model [5]. Indeed, thanks to the beneficial properties of the kernel, such kernel-based regressions allow to construct non-parametric surrogate models, in which the overall complexity in terms of regression unknowns, turns out to be independent form the number of input parameters considered by the model [6]. Moreover, unlike neural network (NN) structures [7], [8], all the aforementioned techniques can be described in terms of a linear model typically estimated from the solution of a convex optimization problem [9], thus leading to several advantages during the model training in terms of training time and convergence with respect to the number of training samples [3].

Despite the above advantages, the main limitation of standard kernel-based regression compared to NN implementations is related to their limited capability to deal with multi-output regression. Unfortunately, such kind of problem are quite common in the modeling of EM structures and components. As an example, we can consider the problem of constructing a surrogate model able to predict the frequency spectra or time waveforms of an EM structure as a function of its internal parameter [7]. In this scenario, unlike multi-output NN structures, the plain application of standard kernel-based regressions turns out to be quite cumbersome, since it would require to build a single-output model for each of the considered frequency/time points, without accounting for possible correlation among them. A possible alternative relies on a two-step procedure in which an unsupervised compression technique, such as the principal component analysis (PCA) [10], is applied on the output dataset to extract a possible inherent correlation existing among several realizations of output data samples at different frequency/time points [3], [11]. Such information is then used to get a compressed representation of the multi-output response, in which the number of actual single-output models required to represent the data can be heavily reduced. It is important to remark that the accuracy of the above modeling scheme is heavily influenced by the capability of the dimensional reduction technique to describe the output training samples and to generalize well on the test data, as well as on the capability of the scalar regression to accurately follow the behavior of the compressed components. Indeed, differently from a pure multiple-output implementation, the effect of a possible small inaccuracy on the compress representation on the vector-valued output is not directly accounted for during the model training.

In order to bridge the gap between the multi-output NN implementation and kernel-based approaches, this work presents a KRR for vector-valued functions, based on the results presented in [12], [13]. The proposed technique is based on the extension of the definition of the reproducing kernel Hilbert space (RKHS) to the case of vector-valued learning problem [13]. The performance of the proposed methodology is investigated on a multi-output problem consisting of predicting the magnitude of the frequency response of a high-speed interconnect as a function of 4 uniform distributed parameters with a "large" variability [3]. The obtained results

are then compared with the ones provided by state-of-the-art techniques such as the combination of the PCA and the LS-SVM and a plain feed-forward multi-output NN structure.

II. Kernel Ridge Regression for Vector-Valued Functions

This section briefly presents the mathematical background behind the proposed KRR for vector-valued functions. Let us start defining the training set $S = \{(\boldsymbol{x}_l, \boldsymbol{y}_l)\}_{l=1}^{L}$, in which $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^P$ is a vector collecting the configurations of the input parameters (e.g., geometrical and electrical parameters of an EM structure) and $\boldsymbol{y}_i = [y_i(f_1), \ldots, y_i(f_D)]^T \in \mathbb{R}^D$ is a vector collecting the corresponding training output (e.g., the frequency samples of a frequency response) for a set of values of the independent variable f_d. The above training set can be rewritten in its compact form as $S = (\mathbf{X}, \mathbf{Y})$ where $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l]^T$ is a $L \times P$ matrix collecting the configurations of the training input and $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l]^T$ is a $L \times D$ matrix associated to the training output. Given the information available in the training set S, our goal is to construct a regression function able to provide an accurate prediction of the output $\boldsymbol{y}(\boldsymbol{x})$ for any configuration of the parameters $\boldsymbol{x} \in \mathcal{X}$. For the proposed vector-valued KRR, such regression function writes:

$$\boldsymbol{y}(\boldsymbol{x}) \approx \sum_{l=1}^{L} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)\boldsymbol{c}_l, \tag{1}$$

where $\mathbf{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^{D \times D}$ is the multi-output kernels function, such as for any pair $\boldsymbol{x}$ and $\boldsymbol{x}'$, $\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')$ is a semi-definite $D \times D$ matrix and $\boldsymbol{c}_l \in \mathbb{R}^D$ are vectors collecting the regression coefficients [12].

Differently from single-output kernel regressions, the above multi-output kernel function jointly acts on both the input space (i.e., on $\boldsymbol{x}$) and on the output dimensions $\{1, \ldots, D\}$. Without loss of generality, this work focuses on a specific type of multi-output kernel called *separable kernels*, such that by fixing both the input pairs $\boldsymbol{x}$ and $\boldsymbol{x}'$, and the output pairs d and d' (i.e., the output pair associated to the frequency components f_d and f'_d), is defined by the following scalar quantity [12], [13]:

$$(\mathbf{K}(\boldsymbol{x}, \boldsymbol{x'}))_{d,d'} = k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x'})k_o(d, d'), \tag{2}$$

where $k_{\boldsymbol{x}}$ and k_o are scalar kernel acting independently on the input space (i.e., $k_{\boldsymbol{x}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$) and output dimensions (i.e., $k_o : \{1, \ldots, D\} \times \{1, \ldots, D\} \rightarrow \mathbb{R}$). In the following results, a RBF kernel is used for both the above scalar kernels, such as:

$$k_{\boldsymbol{x}/o}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp\left(-\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|}{\sigma_{\boldsymbol{x}/o}}\right)^2, \tag{3}$$

where the pair $(\boldsymbol{\theta}, \boldsymbol{\theta}')$ can be any combination of input or output pairs, $\sigma_{\boldsymbol{x}}$ and σ_o are the hyperparameters corresponding to each of the scalar kernel, respectively. In our implementation such parameters are tuned via Bayesian optimization [14] based on a validation set [15].

Extending (2), the kernel matrix associated to all the input training pairs $\mathbf{X}$ and output dimensions writes:

$$\mathbf{K}(\boldsymbol{X}, \boldsymbol{X}) = \mathbf{B} \otimes \mathbf{K}_{\boldsymbol{x}}(\boldsymbol{X}, \boldsymbol{X}), \tag{4}$$

where $\mathbf{B} \in \mathbb{R}^{D \times D}$, such as $(\mathbf{B})_{d,d'} = k_o(d, d')$ and $\mathbf{K}_{\boldsymbol{x}}(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{L \times L}$ such that $(\mathbf{K}_{\boldsymbol{x}}(\boldsymbol{X}, \boldsymbol{X}))_{i,j} = k_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The overall kernel matrix turns out to be a $(DL) \times (DL)$ matrix.

According to [12], the coefficients $\boldsymbol{c}_l$ in (1) can be computed via the solution of the following linear system:

$$\underbrace{(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \lambda N \mathbf{I})}_{\mathbf{A}} \bar{\boldsymbol{c}} - \bar{\boldsymbol{y}} = \mathbf{0}, \tag{5}$$

where $\bar{\boldsymbol{c}} = \mathrm{vec}(\mathbf{C}) \in \mathbb{R}^{LD}$ is a vector collecting all the regression coefficients $\boldsymbol{c}_l$ with $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L]^T \in \mathbb{R}^{L \times D}$, $\bar{\boldsymbol{y}} = \mathrm{vec}(\mathbf{Y})$, λ is a Tikhonov regularizer [15] and $\mathbf{X} \in \mathbb{R}^{L \times P}$ as defined before. Indeed, similar to the scalar case, the regression coefficients $\bar{\boldsymbol{c}}$ can be calculated by solving a linear system of equation, such as:

$$\bar{\boldsymbol{c}} = \mathbf{A}^{-1}\bar{\boldsymbol{y}}. \tag{6}$$

Unfortunately, a plain inversion of the matrix $\mathbf{A}$ can be extremely expansive in terms of both computational time and memory resource, since the computational complexity of matrix inversion scales as $O(n^3)$, where n is the matrix size. This makes the direct inversion algorithm extremely inefficient or intractable in a standard laptop, when the product between the number of training samples (i.e, L) and the output dimensionality D, defining the size of the matrix $\mathbf{A}$, becomes in the order of thousand. To overcome the above limitation, in the proposed algorithm the the linear problem in (5) is solved via an iterative procedure based on the gradient descent algorithm [9], [16]:

$$\bar{\boldsymbol{c}}_k = \bar{\boldsymbol{c}}_{k-1} - \alpha[\mathbf{A}\bar{\boldsymbol{c}}_{k-1} - \bar{\boldsymbol{y}}], \tag{7}$$

where $\bar{\boldsymbol{c}}_k$ represents the unknown regression coefficients estimated at the step k-th and α is a scalar number, known as the learning rate, defining the step-size at each iteration. In particular, the proposed modeling framework implements the conjugate gradient method [16], which provides an efficient version of the above algorithm taylored for semi-definite matrices, as the matrix $\mathbf{A}$. The obtained coefficients can be used within (1), in order to predict the vector-valued output $\boldsymbol{y}$ for any value of the parameters $\boldsymbol{x}$.

It is important to remark that the mathematical framework presented so far describes a preliminary implementation of vector-valued KRR, which has not been optimized in terms of training time and computational complexity. Future work will investigate the beneficial effect of applying compression techniques on the matrix $\boldsymbol{A}$. This will allow to heavily reduce the matrix size and the number of coefficients to be estimated during the model training [9].

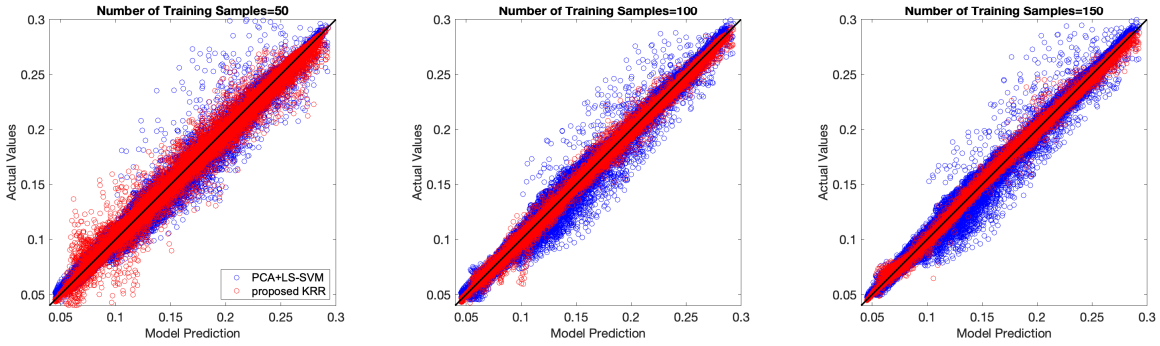

Fig. 1. Comparison of the scatter plots computed from the frequency responses predicted by 3 surrogate model based on the proposed vector-valued KRR and PCA+LS-SVM regression by considering all the frequencies point and the test samples for an increasing number of training samples (i.e., $L = 50$, 100 and 150).

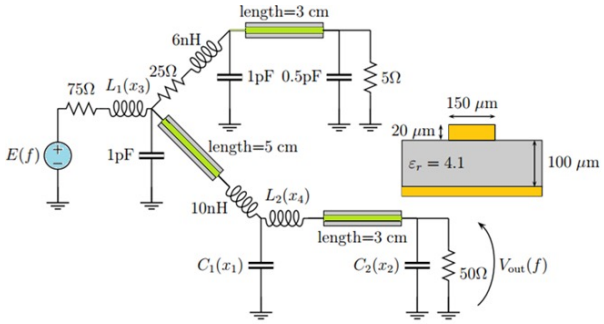

Fig. 2. Network for the application example, reproduced from [3].

III. Numerical Results

The performance of the proposed preliminary implementation of the vector-valued KRR presented in Sec. II, are investigated on the prediction of the parametric behavior of the frequency response $y(\boldsymbol{x}; f) = |V_{out}(f; \boldsymbol{x})|/E(f)|$ of the high-speed link in Fig. 2. The frequency response values are considered in a frequency bandwidth from 1 MHz to 2 GHz, as a function of four normalized parameters collected in the vector $\boldsymbol{x} = [x_1, x_2, x_3, x_4]^T$, defining the values of the lumped components $C_1(x_1)$, $C_2(x_2)$, $L_1(x_3)$, and $L_2(x_4)$ with a uniform variability of $\pm 50\%$ around their mean value (i.e., $\bar{C_1} = 1$ pF, $\bar{C_2} = 0.5$ pF, $\bar{L_1} = 10$ nH and $\bar{L_2} = 10$ nH).

A MATLAB implementation of the above link has been used to generate three different realizations of the training set with $L = 50$, 100 and 150 samples, a validation set with 20 samples and a test set with 1000 samples based on a latin hypercube sampling (LHS) for 100 frequency points (i.e., the output space dimension is $D = 100$). The modeling approach presented in Sec. II has been used to train three different surrogate models by using the available training sets.

Table I compares the performance of the proposed modeling framework, for an increasing number of training samples L, by considering the mean squared error (MSE) computed on the test samples and for all the 100 frequency points. For the sake of completeness, the obtained results are also compared with the ones obtained by two alternative surrogate models based on a two-step scheme combining the PCA and the LS-SVM regression, and a plain feed-forward multi-output NN (additional details are provided in [3]). The results clearly highlight the improved accuracy of the proposed techniques compared with the NN. Moreover, the accuracy provided by the proposed technique seems to outperform also the one obtained by one of the most accurate state-of-the-art kernel-based regression, such as the LS-SVM regression, for which the MSE turns out to be almost constant w.r.t. the number

TABLE I
Comparison of the MSE values computed from the frequency responses predicted by the surrogate models based on the proposed vector-valued KRR, PCA+LS-SVM regression and a simple feed-forward NN structure on all the frequencies points and the test samples for an increasing number of training samples (i.e., $L = 50$, 100 and 150).

Method	MSE $L = 50$	MSE $L = 100$	MSE $L = 150$
NN	8.01×10^{-4}	8.43×10^{-4}	7.12×10^{-4}
PCA+LS-SVM	2.73×10^{-5}	2.22×10^{-5}	2.13×10^{-5}
Proposed KRR	$\mathbf{1.45 \times 10^{-5}}$	$\mathbf{3.56 \times 10^{-6}}$	$\mathbf{2.02 \times 10^{-6}}$

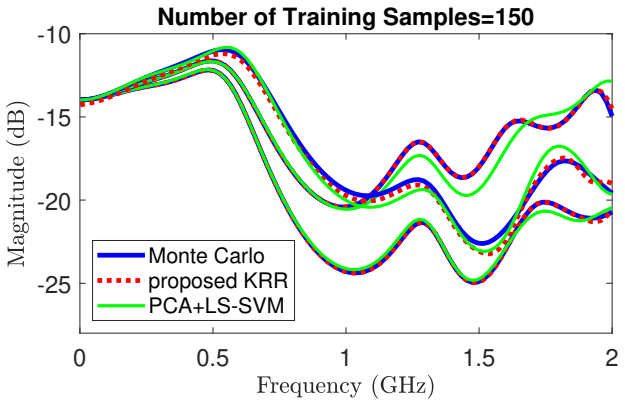

Fig. 3. Parametric plots comparing the frequency responses predicted by the proposed method and the PCA+LS-SVM surrogate models for 3 different realizations of the input parameters.

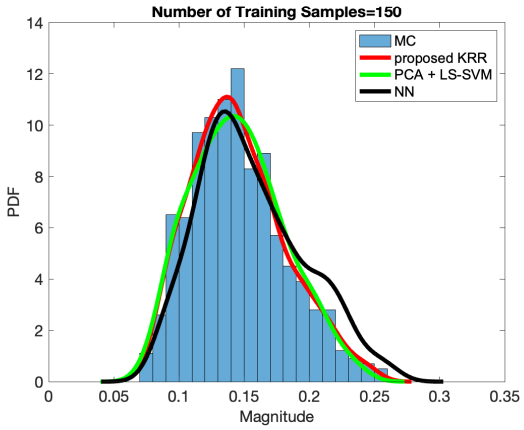

Fig. 4. Comparison of the PDFs computed at the frequency $f = 1.8\,\text{GHz}$ from the predictions of 3 surrogate models based on the proposed vector-valued KRR, PCA+LS-SVM regression and a simple feed-forward NN structure on all the frequencies point and the test samples for $L = 150$ training samples.

of training samples. Such behavior is motivated by the lack of accuracy of the predictions obtained by such technique in the high-frequency range, as shown in the parametric plots of Fig. 3. On the other hand, the improved accuracy of the proposed approach is motivated by the fact that, thanks to the multi-output kernel, the obtained model is able to directly account for the coupling among the components of the multidimensional output, without the need of a two-step procedure based on a direct data compression. As a further proof, Figure 1 shows the scatter plots obtained by considering all the frequency points for the 1000 test samples. It is important to notice that the training cost of the PCA+LS-SVM surrogate is $170\times$ faster than the one of the proposed method.

Moreover, Fig. 4 investigates the effectiveness of the proposed modeling approach within the uncertainty quantification scenario, by comparing the probability density function (PDF) of the output at a single frequency $f = 1.8\,\text{GHz}$ predicted by the considered surrogate models built with $L = 150$ and the corresponding results obtained by a Monte Carlo (MC) simulation with 1000 samples. Also in this case the results clearly highlight the remarkable accuracy of the proposed approach.

IV. Conclusions & Future Works

This paper presented a preliminary implementation of a complete modeling framework based on a vector-value KRR. The proposed approach relies on a generalized definition of the RKHS for vector-valued functions. In the above scenario, the learning problem reduces to the inversion of a *large* linear system. Such computational heavy task has been performed via an iterative algorithm based on the conjugate gradient method. The feasibility and the performance of the proposed approach has been investigated via a parametric and a stochastic analysis on the prediction of the frequency responses of a high-speed link affected by four parameters with a large variability and compared with the ones provided by two state-of-the-art techniques, namely the combination of the PCA with the LS-SVM regression and a plain feed-forward multi-output NN.

It was found that the proposed preliminary version of the multi-output KRR provides the most accurate method among the considered ones, but it also exhibits a large training cost, sine it requires the tuning of a massive number of regression coefficients. Such limitation will be further address in future publications through an advanced and more efficient implementation based on a compressed regression matrix.

References

[1] H. Ma, E. Li, A. C. Cangellaris, and X. Chen, "Support vector regression-based active subspace (SVR-AS) modeling of high-speed links for fast and accurate sensitivity analysis," *IEEE Access*, vol. 8, pp. 74339–74348, 2020.

[2] R. Trinchero, M. Larbi, H. M. Torun, F. G. Canavero, and M. Swaminathan, "Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters," *IEEE Access*, vol. 7, pp. 4056–4066, 2019.

[3] 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.

[4] A. Zanco and S. Grivet-Talocia, "Toward Fully Automated High-Dimensional Parameterized Macromodeling,"" *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 9, pp. 1402-1416, Sept. 2021

[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] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd edition, New York NY: Springer, 1999.

[7] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microwave Magazine*, vol. 11, no. 3, pp. 105–118, 2010.

[8] R. Kumar, S. S. L. Narayan, S. Kumar, S. Roy, B. K. Kaushik, R. Achar, and R. Sharma, "Knowledge-Based Neural Networks for Fast Design Space Exploration of Hybrid Copper-Graphene On-Chip Interconnect Networks," in *Proc. 2021 IEEE Transactions on Electromagnetic Compatibility*, p. 182-195.

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

[10] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., New York, NY, USA: Springer-Verlag, 2002.

[11] 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.

[12] 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.

[13] A. C. Micchelli and M. Pontil, "Kernels for multi-task learning", in *Proc. 2004 17th International Conference on Neural Information Processing Systems*, MIT Press, Cambridge, MA, USA, 921–928.

[14] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[15] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial", arXiv preprint arXiv:1905.12787, 2019.

[16] R. M. Hestenes, E. Stiefel, Eduard, "Methods of Conjugate Gradients for Solving Linear Systems", *Journal of Research of the National Bureau of Standards* 49(6): 409, 1951.