Learning Multidimensional Fourier Series With Tensor Trains

Sander Wahls, Visa Koivunen,
H. Vincent Poor and Michel Verhaegen

This paper has been accepted for publication in the proceedings of the 2nd IEEE Global Conference on Signal and Information Processing (GlobalSIP) which was held in Atlanta, GA, USA in December 2014.

Learning Multidimensional Fourier Series With Tensor Trains

Sander Wahls*, Visa Koivunen†, H. Vincent Poor‡ and Michel Verhaegen*

*Delft Center for Systems and Control, TU Delft, The Netherlands. Email: {s.wahls,m.verhaegen}@tudelft.nl
†Department of Signal Processing and Acoustics, Aalto University, Finland. Email: visa.koivunen@aalto.fi
‡Department of Electrical Engineering, Princeton University, USA. Email: poor@princeton.edu

***Abstract*—How to learn a function from observations of inputs and noisy outputs is a fundamental problem in machine learning. Often, an approximation of the desired function is found by minimizing a risk functional over some function space. The space of candidate functions should contain good approximations of the true function, but it should also be such that the minimization of the risk functional is computationally feasible. In this paper, finite multidimensional Fourier series are used as candidate functions. Their impressive approximative capabilities are illustrated by showing that Gaussian-kernel estimators can be approximated arbitrarily well over any compact set of bandwidths with a fixed number of Fourier coefficients. However, the solution of the associated risk minimization problem is computationally feasible only if the dimension d of the inputs is small because the number of required Fourier coefficients grows exponentially with d. This problem is addressed by using the tensor train format to model the tensor of Fourier coefficients under a low-rank constraint. An algorithm for least-squares regression is derived and the potential of this approach is illustrated in numerical experiments. The computational complexity of the algorithm grows only linearly both with the number of observations N and the input dimension d, making it feasible also for large-scale problems.**

***Index Terms*—Kernels, Risk Minimization, Tensor Train Format, Low-Rank Constraints, Large-Scale Learning**

I. Introduction

Many problems in machine learning can be formulated as risk minimization problems [1], [2]. Consider an unknown function

$$F:\mathbb{R}^d\supset\mathcal{X}\to\mathcal{Y}\subset\mathbb{R},\quad \mathcal{X}:=[-0.5,0.5]^d, \tag{1}$$

which has been sampled at randomly chosen nodes $\mathbf{x}[1],\dots,\mathbf{x}[N]$ under additive noise: $y[i]:=F(\mathbf{x}[i])+\varepsilon[i]$. Let $\mathcal{F}$ denote a space of functions mapping $\mathcal{X}$ to $\mathcal{Y}$ and let $\ell:\mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$ denote a loss function. Then, one wants to minimize the *(frequentist) expected risk*

$$R(f):=\int_{\mathcal{X}\times\mathcal{Y}}\ell\left(f(\mathbf{x}),y\right)dp(\mathbf{x},y),\quad f\in\mathcal{F}.$$

Since the expected risk is often not available, in practice usually the following *regularized empirical risk* is minimized:

$$R_{emp}(f):=\frac{1}{N}\sum_{i=1}^{N}\ell(f(\mathbf{x}[i]),y[i])+\lambda\|f\|_{\mathcal{F}}^2,\ \lambda\geqslant 0,\ f\in\mathcal{F}. \tag{2}$$

Here, $\|f\|_{\mathcal{F}}$ is some norm for functions in $\mathcal{F}$. The term $\lambda\|f\|_{\mathcal{F}}^2$ is often added in order to avoid overfitting to training data samples. Overfitting leads to a reduced generalization capacity for new inputs not included in the training set.

The space of Gaussian-kernel estimators equipped with the norm of the surrounding reproducing kernel Hilbert space [3], [4]

$$\mathcal{G}_{\boldsymbol{\Sigma}}:=\left\{f(\mathbf{x})=\sum_{i=1}^{N}\alpha_i\,\mathrm{e}^{-\|\mathbf{x}[i]-\mathbf{x}\|_{\boldsymbol{\Sigma}}^2}:\alpha_i\in\mathbb{R},\ i=1,\dots,N\right\}, \tag{3}$$

$$\|f\|_{\mathcal{G}_{\boldsymbol{\Sigma}}}^2:=\sum_{i,j=1}^{N}\alpha_i\alpha_j\,\mathrm{e}^{-\|\mathbf{x}[i]-\mathbf{x}[j]\|_{\boldsymbol{\Sigma}}^2},$$

where $\boldsymbol{\Sigma}=\operatorname{diag}(\sigma_1,\dots,\sigma_D)>\mathbf{0}$ is a positive-definite weighting matrix and $\|\mathbf{x}\|_{\boldsymbol{\Sigma}}^2:=\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}$, is a popular example for a space of approximation functions. The space $\mathcal{G}_{\boldsymbol{\Sigma}}$ is N-dimensional, which is why the minimization of the risk (2) becomes infeasible for large-scale data sets. In order to reduce the computational complexity, Rahimi and Recht [5] have proposed to replace $\mathcal{G}_{\boldsymbol{\Sigma}}$ with a lower-dimensional space that is in some sense close to $\mathcal{G}_{\boldsymbol{\Sigma}}$. With samples $\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_D$ taken from a normal distribution with zero mean and covariance $\sqrt{2\boldsymbol{\Sigma}}$, and samples $b_1,\dots,b_D$ taken from the uniform distribution on $[0,2\pi]$, they proposed to replace $\mathcal{G}_{\boldsymbol{\Sigma}}$ with [6, p. 3]

$$\mathcal{R}_{\boldsymbol{\Sigma},D}:=\left\{f(\mathbf{x})=\sum_{i=1}^{D}\alpha_i\cos(\boldsymbol{\omega}_i^T\mathbf{x}+b_i):\alpha_i\in\mathbb{R},\ i=1,\dots,D\right\},$$

$$\|f\|_{\mathcal{R}_{\boldsymbol{\Sigma},D}}^2:=\sum_{i=1}^{D}\alpha_i^2\,\mathrm{e}^{-\|\mathbf{x}[i]\|_{\boldsymbol{\Sigma}}^2}.$$

(Function spaces of this form have already been used earlier [7], [8], but Rahimi and Recht established the connection between the random distribution of the weights and kernels, including approximation guarantees.) In many large-scale learning problems, D can be chosen much smaller than N without sacrificing performance. Another popular approach that can be fit into this framework is based on Nyström sampling [6], [9]. Both approaches, random Fourier features and Nyström sampling, are randomized techniques with probabilistic guarantees of being close to $\mathcal{G}_{\boldsymbol{\Sigma}}$ that hold for a specific, fixed $\boldsymbol{\Sigma}$. See, e.g., [5, Claim 1] or [10, Theorem 3] for examples.

In this paper, we propose finite multidimensional Fourier series as approximation functions. We will demonstrate that finite multidimensional Fourier series are well-suited for function approximation by establishing deterministic guarantees of closeness to $\mathcal{G}_{\boldsymbol{\Sigma}}$. In contrast to the approaches mentioned above, these guarantees are not restricted to a specific choice of $\boldsymbol{\Sigma}$. More precisely, we show that the span of $\bigcup_{\boldsymbol{\Sigma}\in\mathcal{E}}\mathcal{G}_{\boldsymbol{\Sigma}}$, where $\mathcal{E}$ is an arbitrary compact set of positive-definite diagonal weights, can be approximated arbitrarily well through spaces of finite multidimensional Fourier series. As one would expect, this strong property comes at a price. The total number of Fourier coefficients that is needed to achieve these error bounds grows exponentially with the dimension d of the inputs, which seems to render numerical solution of the risk minimization problem infeasible in this case. However, we will demonstrate that the tensor train format [11] can be exploited to beat the curse of dimensionality by applying a low-rank constraint to the tensor of Fourier coefficients.

The paper is structured as follows. In the next section, spaces of finite multidimensional Fourier series are introduced and guarantees of closeness to $\mathcal{G}_{\boldsymbol{\Sigma}}$ are established. Then, in Section III, a numerically efficient algorithm that addresses the minimization of the risk (2) with quadratic loss based on low-rank constraints for the tensor of Fourier coefficients is presented. Numerical experiments are presented afterwards in Section IV. The paper is finally concluded in Section V.

This work was supported in part by the German Research Foundation (DFG) under Grant WA 3139/1-1.

II. Uniform Approximation of Gaussian-Kernel Estimators Through Multidimensional Fourier Series

In this section, first spaces of finite multidimensional Fourier series are introduced. Then, guarantees of closeness between these spaces and the span of $\bigcup_{\boldsymbol{\Sigma}\in\mathcal{E}} \mathcal{G}_{\boldsymbol{\Sigma}}$ are derived.

Let $\mathbf{x} \in \mathcal{X}$ denote an input, and define the index set

$$\mathcal{I} := \left\{-\frac{m}{2}, -\frac{m}{2}+1, \ldots, \frac{m}{2}-1\right\}, \quad m \in 2\mathbb{N}.$$

Then, the *tensor Fourier feature* associated with $\mathbf{x}$ is given by

$$\mathbf{D}(\mathbf{x}) := [d_{\mathbf{l}}(\mathbf{x})]_{\mathbf{l}\in\mathcal{I}^d}, \quad d_{\mathbf{l}}(\mathbf{x}) := \mathrm{e}^{2\pi\,\mathrm{i}\,\mathbf{l}^T\mathbf{x}/p}.$$

The number of indices per dimension $m \in 2\mathbb{N}$ and the scaling factor $p \geqslant 1$ are two fixed parameters that have to be chosen a priori. We propose to use finite multidimensional Fourier series

$$f(\mathbf{x}) = \sum_{\mathbf{l}\in\mathcal{I}^d} c_{\mathbf{l}} \bar{d}_{\mathbf{l}}(\mathbf{x}) =: [\![\mathbf{C}, \mathbf{D}(\mathbf{x})]\!], \tag{4}$$

where $[\![\cdot,\cdot]\!]$ is the tensor dot product and $\mathbf{C} = [c_{\mathbf{l}}]_{\mathbf{l}\in\mathcal{I}^d}$ is a parameter tensor, for risk minimization. The space of all such functions is

$$\mathcal{T}_{m,p} := \left\{ f(\mathbf{x}) = [\![\mathbf{C}, \mathbf{D}(\mathbf{x})]\!] : \mathbf{C} = [c_{\mathbf{l}}]_{\mathbf{l}\in\mathcal{I}^d} \in \mathbb{C}^{\times_{i=1}^d m} \right\}. \tag{5}$$

Using the multidimensional Fourier series of the indicator function $\mathbb{I}_{\mathcal{X}}$ of $\mathcal{X}$ [12, Ch. 8.1], we find that the quadratic norm satisfies

$$\|f\|^2_{\mathcal{T}_{m,p}} := \int_{\mathcal{X}} f(\mathbf{x})\bar{f}(\mathbf{x})d\mathbf{x} = \sum_{\mathbf{l},\mathbf{k}\in\mathcal{I}^d} c_{\mathbf{l}}\bar{c}_{\mathbf{k}} \int_{\mathbb{R}^d} \mathrm{e}^{-2\pi\,\mathrm{i}(\mathbf{k}-\mathbf{l})^T\mathbf{x}/p}$$

$$\times\, \mathbb{I}_{\mathcal{X}}(\mathbf{x})d\mathbf{x} = \sum_{\mathbf{l},\mathbf{k}\in\mathcal{I}^d} c_{\mathbf{l}}\bar{c}_{\mathbf{k}} \prod_{i=1}^d \operatorname{sinc}\left(\frac{k_i - l_i}{p}\right). \tag{6}$$

The following proposition demonstrates that $\mathcal{T}_{m,p}$ provides arbitrarily good approximations of whole families of Gaussian kernels if the parameters m and p are chosen large enough.

Proposition 1. *Let $\epsilon > 0$ and $0 < \underline{\sigma} \leqslant \bar{\sigma}$. Furthermore, denote the set of all $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_d)$ such that $\sigma_i \in [\underline{\sigma}, \bar{\sigma}]$ for all i by $[\underline{\sigma}\mathbf{I}, \bar{\sigma}\mathbf{I}]$. Then, there exist parameters m and p such that for each $\boldsymbol{\Sigma} \in [\underline{\sigma}\mathbf{I}, \bar{\sigma}\mathbf{I}]$ there is a function $f_{\boldsymbol{\Sigma}} \in \mathcal{T}_{m,p}$ that satisfies*

$$\sup_{\mathbf{x},\mathbf{y}\in\mathcal{X}} |\,\mathrm{e}^{-\|\mathbf{x}-\mathbf{y}\|^2_{\boldsymbol{\Sigma}}} - f_{\boldsymbol{\Sigma}}(\mathbf{x}-\mathbf{y})| \leqslant \epsilon. \tag{7}$$

Proof: Please see the appendix. ■

The next corollary extends this result to linear combinations of Gaussians and arbitrary compacts sets of weight matrices.

Corollary 2. *Let $\epsilon > 0$ and let $\mathcal{E} \subset \mathbb{R}^{d\times d}$ denote a compact set of diagonal positive-definite weight matrices. Then, there exist parameters m and p such that for any finite linear combination*

$$g(\mathbf{x}) = \sum_{j=1}^M \beta_j \sum_{i=1}^N \alpha_{i,j}\, \mathrm{e}^{-\|\mathbf{x}[i]-\mathbf{x}\|^2_{\boldsymbol{\Sigma}_j}}, \quad \alpha_{i,j}, \beta_j \in \mathbb{R},\, \boldsymbol{\Sigma}_j \in \mathcal{E},$$

of elements in $\bigcap_{\boldsymbol{\Sigma}\in\mathcal{E}} \mathcal{G}_{\boldsymbol{\Sigma}}$, there exists a function $f \in \mathcal{T}_{m,p}$ that approximates the function g up to the following error:

$$\sup_{\mathbf{x}\in\mathcal{X}} |g(\mathbf{x}) - f(\mathbf{x})| \leqslant \epsilon \sum_{j=1}^M \sum_{i=1}^N |\alpha_{i,j}||\beta_j|. \tag{8}$$

Proof: Since all elements in $\mathcal{E}$ are positive-definite and $\mathcal{E}$ is compact, there exist $0 < \underline{\sigma} \leqslant \bar{\sigma}$ such that $\mathcal{E} \subseteq [\underline{\sigma}\mathbf{I}, \bar{\sigma}\mathbf{I}]$. Proposition 1 thus shows that there are m and p such that, for all i and j,

$$\exists f_{i,j} \in \mathcal{T}_{m,p} : \quad \sup_{\mathbf{x}\in\mathcal{X}} |\,\mathrm{e}^{-\|\mathbf{x}[i]-\mathbf{x}\|^2_{\boldsymbol{\Sigma}_j}} - f_{i,j}(\mathbf{x}[i]-\mathbf{x})| \leqslant \epsilon. \tag{9}$$

The function $f := \sum_{j=1}^M \beta_j \sum_{i=1}^N \alpha_{i,j} f_{i,j}$ belongs to $\mathcal{T}_{m,p}$ because $\mathcal{T}_{m,p}$ forms a vector space. The claim now follows using (9) after the triangle inequality has been applied to $|g(\mathbf{x}) - f(\mathbf{x})|$. ■

III. Breaking The Curse Of Dimensionality

The dimension of the spaces $\mathcal{T}_{m,p}$ defined in (5) corresponds to the number of coefficients in the tensor $\mathbf{C}$, which is $|\mathcal{I}^d| = m^d$. The minimization of the risk (2) for $\mathcal{F} = \mathcal{T}_{m,p}$ is therefore not feasible for high-dimensional inputs. However, motivated by the success of low-rank approximation in the matrix case, in this section an algorithm that addresses the minimization of the risk (2) for the quadratic loss $\ell(x,y) := |x-y|^2$ under a rank-constraint with respect to the tensor train format is proposed. Tensor trains have recently attracted much attention [13] because they combine the advantages of the canonical tensor format (no curse of dimensionality) and the Tucker tensor format (reliable numerical algorithms) [11], [14], [15].

A. Functions In $\mathcal{T}_{m,p}$ With Low-Rank Coefficient Tensors

A tensor $\mathbf{C} = [c_{\mathbf{l}}]_{\mathbf{l}\in\mathcal{I}^d}$ is said to be a *tensor train* of rank at most r if, for any $\mathbf{l} = [l_1, \ldots, l_d]^T \in \mathcal{I}^d$, there exist matrices[1]

$$\mathbf{G}_1(l_1) \in \mathbb{C}^{1\times r},\ \mathbf{G}_2(l_2), \ldots, \mathbf{G}_{d-1}(l_{d-1}) \in \mathbb{C}^{r\times r},\ \mathbf{G}(l_d) \in \mathbb{C}^{r\times 1}$$

such that $c_{\mathbf{l}} = \mathbf{G}_1(l_1) \times \cdots \times \mathbf{G}_d(l_d)$ [11]. We denote the set of all such tensor trains by $\mathfrak{T}^r_m$. The corresponding subset of $\mathcal{T}_{m,p}$ is

$$\mathcal{T}^r_{m,p} := \{ f(\mathbf{x}) = [\![\mathbf{C}, \mathbf{D}(\mathbf{x})]\!] : \mathbf{C} \in \mathfrak{T}^r_m \}.$$

At this point, note that $\mathbf{D}(\mathbf{x})$ is a tensor train of rank one because $d_{\mathbf{l}}(\mathbf{x}) = \mathrm{e}^{2\pi\,\mathrm{i}\,l_1 x_1/p} \times \cdots \times \mathrm{e}^{2\pi\,\mathrm{i}\,l_d x_d/p}$. The next lemma thus shows how functions in $\mathcal{T}^r_{m,p}$ can be evaluated efficiently.

Lemma 3 (In part from [11], p. 2309)**.** *Consider two tensor trains*

$$\begin{aligned} \mathbf{C} &= [c_{\mathbf{l}}]_{\mathbf{l}\in\mathcal{I}^d} \in \mathfrak{T}^r_{m,p}, \quad c_{\mathbf{l}} = \mathbf{G}_1(l_1) \times \cdots \times \mathbf{G}_d(l_d), \\ \mathbf{Z} &= [z_{\mathbf{l}}]_{\mathbf{l}\in\mathcal{I}^d} \in \mathfrak{T}^1_{m,p}, \quad z_{\mathbf{l}} = z_1(l_1) \times \cdots \times z_d(l_d), \end{aligned} \tag{10}$$

and define matrices $\boldsymbol{\Gamma}_k(\mathbf{Z}) := \sum_{l\in\mathcal{I}} \mathbf{G}_k(l)\bar{z}_k(l)$,

$$\mathbf{L}_k(\mathbf{Z}) := \begin{cases} \boldsymbol{\Gamma}_1(\mathbf{Z}) \times \cdots \times \boldsymbol{\Gamma}_{k-1}(\mathbf{Z}) & ,k > 1 \\ 1 & ,k = 1 \end{cases},$$

$$\mathbf{R}_k(\mathbf{Z}) := \begin{cases} \boldsymbol{\Gamma}_{k+1}(\mathbf{Z}) \times \cdots \times \boldsymbol{\Gamma}_d(\mathbf{Z}) & ,k < d \\ 1 & ,k = d \end{cases},$$

$$\mathbf{H}_k(\mathbf{Z}) := \begin{cases} \begin{bmatrix} \bar{z}_k(-\frac{m}{2}) & \ldots & \bar{z}_k(\frac{m}{2}-1) \end{bmatrix} \otimes \mathbf{I}_r & ,k > 1 \\ \begin{bmatrix} \bar{z}_k(-\frac{m}{2}) & \ldots & \bar{z}_k(\frac{m}{2}-1) \end{bmatrix} & ,k = 1 \end{cases},$$

$$\mathbf{G}^L_k := \begin{bmatrix} \mathbf{G}_k(-\frac{m}{2})^T & \ldots & \mathbf{G}_k(\frac{m}{2}-1)^T \end{bmatrix}^T, \tag{11}$$

where $\otimes$ denotes the Kronecker product [16] and $\mathbf{I}_r$ denotes the $r \times r$ identity matrix. Then, with $\operatorname{vec}(\cdot)$ being the operator that maps matrices to vectors by stacking their columns,

$$[\![\mathbf{C}, \mathbf{Z}]\!] = \Big(\mathbf{R}^T_k(\mathbf{Z}) \otimes \mathbf{L}_k(\mathbf{Z})\mathbf{H}_k(\mathbf{Z})\Big) \operatorname{vec}(\mathbf{G}^L_k). \tag{12}$$

Proof: By definition, $\boldsymbol{\Gamma}_k = \mathbf{H}_k\mathbf{G}^L_k$. The Kronecker product has the well-known property that $\boldsymbol{\mathcal{A}}\mathbf{X}\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{C}}$ for arbitrary matrices $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}$ and $\mathbf{X}$ of compatible dimensions if and only if $(\boldsymbol{\mathcal{B}}^T \otimes \boldsymbol{\mathcal{A}}) \operatorname{vec}(\mathbf{X}) = \operatorname{vec}(\boldsymbol{\mathcal{C}})$ [16]. Thus, with $\boldsymbol{\mathcal{B}} = \mathbf{I}_r$, one obtains $\operatorname{vec}(\boldsymbol{\Gamma}_k) = (\mathbf{I}_r \otimes \mathbf{H}_k) \operatorname{vec}(\mathbf{G}^L_k)$. We find that

$$[\![\mathbf{C}, \mathbf{Z}]\!] = \sum_{l_1\in\mathcal{I}} \cdots \sum_{l_d\in\mathcal{I}} \mathbf{G}_1(l_1)\bar{z}_1(l_1) \times \cdots \times \mathbf{G}_d(l_d)\bar{z}_d(l_d)$$

[1] The definition of a tensor train given here has been simplified [11].

$$=\boldsymbol{\Gamma}_1 \times \cdots \times \boldsymbol{\Gamma}_d = \mathbf{L}_k\boldsymbol{\Gamma}_k\mathbf{R}_k = \left(\mathbf{R}_k^T \otimes \mathbf{L}_k\right)\operatorname{vec}(\boldsymbol{\Gamma}_k)$$
$$=\left(\mathbf{R}_k^T \otimes \mathbf{L}_k\right)(\mathbf{I}_r \otimes \mathbf{H}_k)\operatorname{vec}(\mathbf{G}_k^L) = \text{RHS of (12).} \qquad \blacksquare$$

The norm of a function in $\mathcal{T}_{m,p}^r$ can be computed as follows.

Lemma 4. *Let* $\mathbf{C}$ *be a tensor train as in (10), and let* $\mathbf{A}_k$ *and* $\mathbf{B}_k$ *denote matrices that satisfy (14) and (15) on top of the next page. Furthermore, let* $\mathbf{S}$ *be such that* $\mathbf{S}^*\mathbf{S} = [\operatorname{sinc}(\frac{l-s}{p})]_{l,s\in\mathcal{I}}$. *Then, with* $\mathbf{G}_k^L$ *as in (11), the norm of* $f(\mathbf{x}) = [\![\mathbf{C},\mathbf{D}(\mathbf{x})]\!]$ *is given by*

$$\|f\|_{\mathcal{T}_{m,p}}^2 = \left\|\left(\mathbf{A}_k^T \otimes \mathbf{S} \otimes \mathbf{B}_k\right)\operatorname{vec}(\mathbf{G}_k^L)\right\|^2. \qquad (13)$$

Proof: Using (6) and [17, p. 294], one obtains

$$\|f\|^2 = \operatorname{tr}\Big\{\sum_{l_1,s_1\in\mathcal{I}}\cdots\sum_{l_d,s_d\in\mathcal{I}} \mathbf{G}_d(s_d)^* \times \cdots \times \mathbf{G}_1(s_1)^*$$
$$\times\, \mathbf{G}_1(l_1) \times \cdots \times \mathbf{G}_d(l_d) \prod_{i=1}^{d} \operatorname{sinc}\left(\frac{s_i - l_i}{p}\right)\Big\}$$
$$= \operatorname{tr}\Big\{\sum_{l_k,s_k\in\mathcal{I}} \mathbf{A}_k^*\mathbf{G}_k(s_k)^*\mathbf{B}_k^*\mathbf{B}_k\mathbf{G}_k(l_k)\mathbf{A}_k \operatorname{sinc}\left(\frac{s_k - l_k}{p}\right)\Big\}$$
$$= \operatorname{tr}\Big\{\sum_{s\in\mathcal{I}} \mathbf{A}_k^*\mathbf{G}_k(s)^* \sum_{l\in\mathcal{I}} \operatorname{sinc}\left(\frac{s-l}{p}\right)\mathbf{B}_k^*\mathbf{B}_k\mathbf{G}_k(l)\mathbf{A}_k\Big\}$$
$$= \operatorname{tr}\Big\{\mathbf{A}_k^*(\mathbf{G}_k^L)^*\left(\mathbf{S}^*\mathbf{S}\otimes\mathbf{B}_k^*\mathbf{B}_k\right)\mathbf{G}_k^L\mathbf{A}_k\Big\}$$
$$= \left\|(\mathbf{S}\otimes\mathbf{B}_k)\mathbf{G}_k^L\mathbf{A}_k\right\|_F^2 = \text{RHS of (13).} \qquad \blacksquare$$

Remark 5. The matrix $[\operatorname{sinc}(\frac{l-s}{p})]_{l,s}$ is positive semi-definite by (6).

Remark 6. The matrices $\boldsymbol{\Gamma}_1,\ldots,\boldsymbol{\Gamma}_d$ in Lemma 3 can be computed using $\mathcal{O}(dmr^2)$ floating point operations (*flops*). Forming $\mathbf{L}_k$ and $\mathbf{R}_k$ then takes $\mathcal{O}(dr^2)$ flops because $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_d$ are vectors. The computation of $\mathbf{A}_k$ and $\mathbf{B}_k$ in Lemma 4 requires $\mathcal{O}(dm^2r^3)$ flops.

B. Risk Minimization Over Low-rank Coefficient Tensors

The risk (2) is in general not convex over $\mathcal{F} = \mathcal{T}_{m,p}^r$. We propose to use an alternating least squares approach as in [14] and [15] to find a local minimum. For the quadratic loss $\ell(x,y) = |x-y|^2$, the risk (2) can be rewritten as follows. Let $f(\mathbf{x}) = [\![\mathbf{C},\mathbf{D}(\mathbf{x})]\!]$, where the coefficient tensor train $\mathbf{C} = [c_\mathbf{l}]$ is given by $c_\mathbf{l} = \mathbf{G}_1(l_1)\times\cdots\times\mathbf{G}_d(l_d)$. Then, for any k and with $\mathbf{Z}_i := \mathbf{D}(\mathbf{x}[i])$, the Lemmas 3 and 4 show that the risk can be written as

$$R_{emp}(f) = \frac{1}{N}\sum_{j=1}^{N}|y[j] - [\![\mathbf{C},\mathbf{D}(\mathbf{x}[j])]\!]|^2 + \lambda\|f\|_{\mathcal{T}_{m,p}}^2 = \frac{1}{N}\times$$
$$\left\|\begin{bmatrix} y[1] \\ \vdots \\ y[N] \\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{R}_k(\mathbf{Z}_1)^T\otimes\mathbf{L}_k(\mathbf{Z}_1)\mathbf{H}_k(\mathbf{Z}_1) \\ \vdots \\ \mathbf{R}_k(\mathbf{Z}_N)^T\otimes\mathbf{L}_k(\mathbf{Z}_N)\mathbf{H}_k(\mathbf{Z}_N) \\ \sqrt{N\lambda}\mathbf{A}_k^T\otimes\mathbf{S}\otimes\mathbf{B}_k\end{bmatrix}\operatorname{vec}(\mathbf{G}_k^L)\right\|^2. \qquad (16)$$

The size of the coefficient matrix in (16) is $(N+mr^2)\times mr^2$ for $k \neq 1, d$. Thus, a single *core* $\{\mathbf{G}_k(l)\}_{l\in\mathcal{I}}$ of the tensor $\mathbf{C}$ can be updated by solving the linear least squares problem to minimize (16). In the alternating least squares approach, the cores are updated sequentially. In one iteration of the algorithm, first $\mathbf{G}_1$ is updated by minimizing (16), then $\mathbf{G}_2$ is updated in the same way, etc., until $\mathbf{G}_d$ has been updated. The iterations are repeated until convergence.

An important implementation detail arises because the representation $c_\mathbf{l} = \mathbf{G}_1(l_1)\times\cdots\times\mathbf{G}_d(l_d)$ of a tensor train is highly non-unique. To avoid numerical problems, the tensor train should be stored using a canonical representation. A representation $c_\mathbf{l} = \mathbf{G}_1(l_1)\times\cdots\times\mathbf{G}_d(l_d)$ is *left-orthonormal*, if the left-unfolding $\mathbf{G}_k^L$ defined in (11) of each but the last core has orthonormal rows (or columns). A common approach to ensure that the tensor train is left-orthonormal after each iteration is to use a QR factorization to make the updated core left-orthonormal by replacing its left-unfolding with $\mathbf{Q}$. The tensor train remains the same if the non-orthonormal part that corresponds to $\mathbf{R}$ is shifted into the next core. However, since that core will be overwritten in the next step, this is not necessary. The left-unfolding $\mathbf{G}_k^L$ can instead be directly orthonormalized using, e.g., the modified Gram-Schmidt method. Algorithm 1 provides an overview of the procedure.

Algorithm 1 Alternating minimization of R_{emp} for $\mathcal{F} = \mathcal{T}_{m,p}^r$

```
Input: Left-orthonormal initial guess $c_\mathbf{l} = \mathbf{G}_1(l_1)\times\cdots\times\mathbf{G}_d(l_d)$
Repeat until convergence:
for $k = 1,\ldots,d$:
  • Compute $\{\mathbf{L}_k(\mathbf{Z}_i)\}_{i=1}^N$, $\{\mathbf{R}_k(\mathbf{Z}_i)\}_{i=1}^N$, $\mathbf{A}_k$ and $\mathbf{B}_k$
  • Update $\{\mathbf{G}_k(l)\}_{l\in\mathcal{I}}$ by minimizing (16) and using (11)
  • If $k < d$: orthonormalize $\{\mathbf{G}_k(l)\}_{l\in\mathcal{I}}$
```

Remark 7. Algorithm 1 converges locally around a local minimum if the Hessian of this minimum has maximal rank [15, Corollary 2.9].

Remark 8. The minimization of (16) in a least-squares sense requires $\mathcal{O}((N+mr^2)m^2r^4)$ flops using standard techniques. The orthonormalization of a core requires $\mathcal{O}(mr^3)$ flops. Remark 6 implies that forming the coefficient matrix in (16) in general requires $\mathcal{O}(d(N+mr)mr^2)$ flops. In Algorithm 1 however, where the cores are updated sequentially, it is possible to do this more efficiently. The matrices $\mathbf{L}_k$ can be updated as $\mathbf{L}_k = \mathbf{L}_{k-1}\boldsymbol{\Gamma}_{k-1}$. The matrices $\mathbf{R}_1,\ldots,\mathbf{R}_d$ can be efficiently precomputed at the beginning of each iteration (when $k=1$) using the formula $\mathbf{R}_{j-1} = \boldsymbol{\Gamma}_j\mathbf{R}_j$ because the $\mathbf{R}_{k+1},\ldots,\mathbf{R}_d$ are independent of $\boldsymbol{\Gamma}_k$. A similar strategy may be used to cope with the regularization matrices $\mathbf{A}_k$ and $\mathbf{B}_k$. In this way, the costs of finding the coefficient matrix in (16) can be reduced to $\mathcal{O}((N+mr)mr^2)$ flops. Then, the total cost of updating one core is $\mathcal{O}((N+mr^2)m^2r^4)$ flops, and a complete iteration in Algorithm 1 can be carried out using only $\mathcal{O}(d(N+mr^2)m^2r^4)$ flops.

IV. Numerical Experiments

Setup: We have benchmarked Algorithm 1 for minimizing (2) with $\mathcal{F} = \mathcal{T}_{m,p}^r$ and $\ell(x,y) = |x-y|^2$ against standard *kernel ridge regression* [21] (*KRR*, $\mathcal{F} = \mathcal{G}_{\boldsymbol{\Sigma}}$) and random Fourier features (*RFF*, $\mathcal{F} = \mathcal{R}_{\boldsymbol{\Sigma},D}$) for several data sets that have been downloaded from [22]. Each data set has first been been randomly permuted and then partitioned into a training data set (70% of the data) and a testing data set (30% of the data). Parameters which are not given in Figure 1 have been chosen by performing a grid search. Each combination of the parameters was evaluated by performing a 5-fold cross validation on the training data. The predictors with respect to the best parameters were then trained on the training data and evaluated on the test data. The reported errors are average values taken over 10 experiments.

Implementation Details: The inputs $\mathbf{x}[1],\ldots,\mathbf{x}[N]$ have been rescaled (all with the same scalar) such that $\mathbf{x}[1],\ldots,\mathbf{x}[N]\in\mathcal{X}$ with $\mathcal{X}$ as in (1). Uniform weights $\boldsymbol{\Sigma} = \sigma\mathbf{I}$ have been used in order to keep the grid search feasible. The dimension of the random Fourier features was chosen equal to the number of floats needed to store the tensor train used in Algorithm 1: $D = 2mr + (d-2)mr^2$. When random initializations were used (Alg. 1 and RFF), three different initializations have been evaluated and the one with the smallest training error was used. Algorithm 1 always performed 10 iterations. The source code is available online at http://bitbucket.com/wahls/mdfourier.

$$\mathbf{A}_k\mathbf{A}_k^* = \sum_{l_{k+1},s_{k+1}\in\mathcal{I}} \mathbf{G}_{k+1}(l_{k+1}) \left(\cdots \left(\sum_{l_d,s_d\in\mathcal{I}} \mathbf{G}_d(l_d)\mathbf{G}_d(s_d)^* \operatorname{sinc}\left(\frac{s_d-l_d}{p}\right)\right)\cdots\right) \mathbf{G}_{k+1}(s_{k+1})^* \operatorname{sinc}\left(\frac{s_{k+1}-l_{k+1}}{p}\right), \tag{14}$$

$$\mathbf{B}_k^*\mathbf{B}_k = \sum_{l_{k-1},s_{k-1}\in\mathcal{I}} \mathbf{G}_{k-1}(s_{k-1})^* \left(\cdots \left(\sum_{l_1,s_1\in\mathcal{I}} \mathbf{G}_1(s_1)^*\mathbf{G}_1(l_1) \operatorname{sinc}\left(\frac{s_1-l_1}{p}\right)\right)\cdots\right) \mathbf{G}_{k-1}(l_{k-1}) \operatorname{sinc}\left(\frac{s_{k-1}-l_{k-1}}{p}\right). \tag{15}$$

			Alg. 1				**KRR**		**RFF**		
data set	N	d	**Test err.**	CV err.	m	r	**Test err.**	CV err.	**Test err.**	CV err.	D
airfoil [18]	1503	5	**0.03**	0.03	12	2	**0.03**	0.03	0.10	0.03	192
yacht [19]	308	6	0.06	0.05	6	1	**0.04**	0.05	0.26	0.25	36
concrete [20]	1030	8	**0.15**	0.14	8	3	**0.15**	0.15	0.16	0.16	480

Figure 1. Test and cross-validation errors $\|\mathbf{y}-\hat{\mathbf{y}}\|/\|\mathbf{y}\|$. The vector $\mathbf{y}$ contains the stacked outputs, $\hat{\mathbf{y}}$ predicted values. Algorithm 1 performs very close to kernel ridge regression (KRR) and better than random Fourier features with the same amount of memory (RFF) on all three data sets.

Results: The results are reported in Figure 1. Algorithm 1 has been able to perform similarly to kernel ridge regression with less resources (D floats instead of N) on all three data sets. Algorithm 1 performed better than random Fourier features that have been provided the same amount of memory in all cases. On the airfoil and yacht data sets, the test error could be reduced significantly by a factor of three and five, respectively. The high average test error for random Fourier features on the airfoil data set was caused by a single experiment (out of ten).

V. Conclusion

The numerical experiments have confirmed the approximative capabilities of multidimensional Fourier series even if a low-rank constraint is placed on the tensor of Fourier coefficients. The proposed algorithm performs as well as kernel ridge regression and better than random Fourier features – often significantly – while having low computational costs. An improved alternating least squares method as in [14] that adapts the dimensions of the cores of the tensor train during the iterations would be an interesting topic for future research.

Appendix: Proof of Proposition 1

Lemma 9 ([23])**.** *Let $m \in 2\mathbb{N}$, $p \geqslant 1$, $\sigma > 0$, and define $E_1(p,\sigma) := 2\,\mathrm{e}^{-\frac{\sigma(2p-1)^2}{4}}\left(1+\frac{1}{\sigma p(2p-1)}\right)$ and $E_2(m,p,\sigma) := \frac{\sqrt{\pi}}{p\sqrt{\sigma}}\,\mathrm{e}^{-\frac{m^2\pi^2}{4p^2\sigma}}\left(1+\frac{2\sigma p^2}{m\pi^2}\right)$. Then, with $b_l(p,\sigma) := \frac{\sqrt{\pi}}{p\sqrt{\sigma}}\,\mathrm{e}^{-l^2\pi^2/(\sigma p^2)}$,*

$$\sup_{\substack{x\in\mathbb{R}\\|x|\leqslant\frac{1}{2}}} |\,\mathrm{e}^{-\sigma x^2} - \textstyle\sum_{l=-\frac{m}{2}}^{\frac{m}{2}-1} b_l(p,\sigma)\,\mathrm{e}^{-2\pi\,\mathrm{i}\,lx/p}\,| \leqslant E_1 + E_2.$$

Lemma 10. *Let $\epsilon > 0$ and $0 < \underline{\sigma} \leqslant \bar{\sigma}$. Then, there are m and p s.t.*

$$\forall\sigma\in[\underline{\sigma},\bar{\sigma}]: \quad \sup_{\substack{x\in\mathbb{R}\\|x|\leqslant\frac{1}{2}}} |\,\mathrm{e}^{-\sigma x^2} - \textstyle\sum_{l=-\frac{m}{2}}^{\frac{m}{2}-1} b_l(p,\sigma)\,\mathrm{e}^{-2\pi\,\mathrm{i}\,lx/p}\,| \leqslant \epsilon.$$

Proof: First, choose $p \geqslant 1$ large enough such that $\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\exp\left(-\sigma(2p-1)^2/4\right) = \exp\left(-\underline{\sigma}(2p-1)^2/4\right) \leqslant \frac{\varepsilon}{8}$ and $\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]} 1/(\sigma p(2p-1)) = 1/(\underline{\sigma}p(2p-1)) \leqslant 1$. Then,

$$\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]} E_1(p,\sigma) \leqslant \left(2\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\mathrm{e}^{-\frac{\sigma(2p-1)^2}{4}}\right) \times\left(1+\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\frac{1}{\sigma p(2p-1)}\right) = \frac{\varepsilon}{2}. \tag{17}$$

Next, choose $m > 0$ large enough such that

$$\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\mathrm{e}^{-\frac{m^2\pi^2}{4p^2\sigma}} = \mathrm{e}^{-\frac{m^2\pi^2}{4p^2\bar{\sigma}}} \leqslant \frac{\varepsilon}{4}\sqrt{\frac{\sigma}{\pi}},$$

$$\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\frac{\sqrt{\pi}}{p\sqrt{\sigma}}\left(1+\frac{2\sigma p^2}{m\pi^2}\right) \leqslant \sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\frac{\sqrt{\pi}}{\sqrt{\sigma}}\left(1+\frac{2\sigma p^2}{m\pi^2}\right) \leqslant\sqrt{\frac{\pi}{\underline{\sigma}}}\left(1+\frac{2\bar{\sigma}p^2}{m\pi^2}\right) \leqslant 2\sqrt{\frac{\pi}{\underline{\sigma}}};$$

$$\Longrightarrow \sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]} E_2(m,p,\sigma) \leqslant \left(\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\mathrm{e}^{-\frac{m^2\pi^2}{4p^2\sigma}}\right) \times\left(\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\frac{\sqrt{\pi}}{p\sqrt{\sigma}}\left(1+\frac{2\sigma p^2}{m\pi^2}\right)\right) = \frac{\epsilon}{2}. \tag{18}$$

The claim now follows from Lemma 9, (17) and (18):

$$\sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]}\sup_{\substack{x\in\mathbb{R}\\|x|\leqslant\frac{1}{2}}} |\,\mathrm{e}^{-\sigma x^2} - \textstyle\sum_{l=-\frac{m}{2}}^{\frac{m}{2}-1} b_l\,\mathrm{e}^{-2\pi\,\mathrm{i}\,lx/p}\,| \leqslant \sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]} E_1(p,\sigma) + \sup_{\sigma\in[\underline{\sigma},\bar{\sigma}]} E_2(m,p,\sigma) \leqslant \frac{\epsilon}{2}+\frac{\epsilon}{2}. \qquad \blacksquare$$

Proof of Proposition 1: Since any rescaling of the inputs can be compensated by changing $\mathbf{\Sigma}$, we assume $\mathcal{X} = [-\frac{1}{4},\frac{1}{4}]^d$ during this proof without loss of generality. We only discuss the case $d = 2$.

Lemma 10 shows that there are m and p such that the functions $g_\sigma(x) := \sum_{l=-\frac{m}{2}}^{\frac{m}{2}-1} b_l(p,\sigma)\,\mathrm{e}^{2\pi\,\mathrm{i}\,lx/p}$ satisfy

$$\forall\sigma\in[\underline{\sigma},\bar{\sigma}]: \quad \sup_{|x|,|y|\leqslant\frac{1}{4}} |\,\mathrm{e}^{-\sigma(x-y)^2} - g_\sigma(x-y)| \leqslant \sqrt{\epsilon}/2. \tag{19}$$

Let us fix an arbitrary $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1,\sigma_2) \in [\underline{\sigma}\mathbf{I},\bar{\sigma}\mathbf{I}]$. We set $\mathbf{C}_{\mathbf{\Sigma}} := [b_{l_1}(p,\sigma_1)b_{l_2}(p,\sigma_2)]_{\mathbf{l}=[l_1,l_2]^T\in\mathcal{I}^2}$, and define a function $f_{\mathbf{\Sigma}} \in \mathcal{T}_{m,p}$ by $f_{\mathbf{\Sigma}}(\mathbf{x}) := [\![\mathbf{C}_{\mathbf{\Sigma}},\mathbf{D}(\mathbf{x})]\!]$. This function is a product of two g_σ:

$$f_{\mathbf{\Sigma}}(\mathbf{x}) = \textstyle\sum_{\mathbf{l}\in\mathcal{I}^2} b_{l_1}b_{l_2}\,\mathrm{e}^{2\pi\,\mathrm{i}\,\mathbf{l}^T\mathbf{x}/p} = \left(\sum_{l_1=-\frac{m}{2}}^{\frac{m}{2}-1} b_{l_1}\,\mathrm{e}^{2\pi\,\mathrm{i}\,l_1x_1/p}\right) \times\left(\sum_{l_2=-\frac{m}{2}}^{\frac{m}{2}-1} b_{l_2}\,\mathrm{e}^{2\pi\,\mathrm{i}\,l_2x_2/p}\right) = g_{\sigma_1}(x_1)g_{\sigma_2}(x_2).$$

Let us now fix arbitrary $\mathbf{x},\mathbf{y} \in \mathcal{X}$, and define $e_i(\mathbf{x},\mathbf{y},\mathbf{\Sigma}) := \mathrm{e}^{-\sigma_i(x_i-y_i)^2} - g_{\sigma_i}(x_i-y_i)$. Note that $|e_i| \leqslant \sqrt{\epsilon}/2$ by (19). The claim (7) now follows from

$$\begin{aligned}
&\left|\mathrm{e}^{-\|\mathbf{x}-\mathbf{y}\|_{\mathbf{\Sigma}}^2} - f_{\mathbf{\Sigma}}(\mathbf{x}-\mathbf{y})\right| \\
&= \left|\mathrm{e}^{-\sigma_1(x_1-y_1)^2}\,\mathrm{e}^{-\sigma_2(x_2-y_2)^2} - g_{\sigma_1}(x_1-y_1)g_{\sigma_2}(x_2-y_2)\right| \\
&= \left|\mathrm{e}^{-\sigma_1(x_1-y_1)^2} e_2 + e_1\,\mathrm{e}^{-\sigma_2(x_2-y_2)^2} - e_1e_2\right| \\
&\leqslant |\,\mathrm{e}^{-\sigma_1(x_1-y_1)^2} e_2| + |e_1\,\mathrm{e}^{-\sigma_2(x_2-y_2)^2}\,| + |e_1e_2| \\
&\leqslant |e_1| + |e_1e_2| + |e_2| \leqslant (|e_1|+|e_2|)^2 \leqslant (2\sqrt{\epsilon}/2)^2 = \epsilon.
\end{aligned}$$

$\blacksquare$

References

[1] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[2] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, Mar. 2014.

[3] P. Exterkate, "Model selection in kernel ridge regression," *Comput. Stat. Data An.*, vol. 68, pp. 1–16, Dec. 2013.

[4] J. Hainmueller and C. Hazlett, "Kernel regularized least squares: Reducing misspecification bias with a flexible and interpretable machine learning approach," *Political Analysis*, vol. 22, no. 2, pp. 143–168, 2014.

[5] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Ann. Conf. Neural Inf. Process. Syst. (NIPS)*, Vancouver, Canada, Dec. 2007.

[6] T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou, "Nyström method vs random Fourier features: A theoretical and empirical comparison," in *Proc. Ann. Conf. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, Dec. 2012.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*, Budapest, Hungary, Jul. 2004.

[8] ——, "Extreme learning machine: Theory and applications," *Neurocomput.*, vol. 70, no. 1–3, pp. 489–501, 2006.

[9] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, "Recent advances of large-scale linear classification," *Proc. IEEE*, vol. 100, no. 9, pp. 2584–2603, Sep. 2012.

[10] P. Drineas and M. W. Mahoney, "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *J. Mach. Learn. Res.*, vol. 6, pp. 2153–2175, 2005.

[11] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[12] B. Osgood, "The Fourier transform and its applications," Stanford Univ., Lect. Not. EE 261, Fall 2007, http://see.stanford.edu/see/materials/lsoftaee261/handouts.aspx.

[13] L. Grasedyck, D. Kressner, and C. Tobler, "A literature survey of low-rank tensor approximation techniques," *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 53–78, Aug. 2013.

[14] S. Holtz, T. Rohwedder, and R. Schneider, "The alternating linear scheme for tensor optimization in the tensor train format," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. A683–A713, 2012.

[15] T. Rohwedder and A. Uschmajew, "On local convergence of alternating schemes for optimization of convex problems in the tensor train format," *SIAM J. Numer. Anal.*, vol. 51, no. 2, pp. 1134–1162, 2013.

[16] H. V. Henderson and S. R. Searle, "The vec-permutation matrix, the vec operator and Kronecker products: A review," *Linear Multilinear Algebra*, vol. 9, no. 4, pp. 271 – 288, 1981.

[17] C. Van Loan and N. P. Pitsianis, "Approximation with Kronecker products," in *Linear Algebra for Large Scale and Real Time Applications*, M. S. Moonen, G. H. Golub, and B. R. L. De Moor, Eds. Dordrecht, Netherlands: Kluwer Pub., 1993, pp. 293–314.

[18] T. F. Brooks, D. S. Pope, and M. A. Marcolini, "Airfoil self-noise and prediction," NASA, Tech. Rep. RP-1218, 1989.

[19] J. Gerritsma, R. Onnink, and A. Versluis, "Geometry, resistance and stability of the Delft systematic yacht hull series," *Int. Shipbuilding Progr.*, vol. 28, pp. 276–297, 1981.

[20] I.-C. Yeh, "Modeling of strength of high-performance concrete using artificial neural networks," *Cem. Concr. Res.*, vol. 28, no. 12, pp. 1797–1808, Dec. 1998.

[21] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proc. Int. Conf. Mach. Learning (ICML)*, Madison, WI, Jul. 1998.

[22] K. Bache and M. Lichman, "UCI machine learning repository," UC Irvine, School of Inf. Comput. Sci., 2014, http://archive.ics.uci.edu/ml.

[23] S. Kunis, D. Potts, and G. Steidl, "Fast Gauss transforms with complex parameters using NFFTs," *J. Numer. Math.*, vol. 14, pp. 295–303, 2006.