INTERPRETABLE APPROXIMATION OF HIGH-DIMENSIONAL DATA

A PREPRINT

Daniel Potts
Faculty of Mathematics
Chemnitz University of Technology
09107 Chemnitz
potts@math.tu-chemnitz.de

Michael Schmischke
Faculty of Mathematics
Chemnitz University of Technology
09107 Chemnitz
michael.schmischke@math.tu-chemnitz.de

ABSTRACT

In this paper we apply the previously introduced approximation method based on the ANOVA (analysis of variance) decomposition and Grouped Transformations to synthetic and real data. The advantage of this method is the interpretability of the approximation, i.e., the ability to rank the importance of the attribute interactions or the variable couplings. Moreover, we are able to generate an attribute ranking to identify unimportant variables and reduce the dimensionality of the problem. We compare the method to other approaches on publicly available benchmark datasets.

Keywords ANOVA · high-dimensional · approximation · intrepretability · fast Fourier methods

1 Introduction

Building models for creating predictions based on empirical data is a current and active research topic with numerous applications. The amount of data being collected is ever-increasing resulting in high-dimensional datasets with corresponding regression or classification problems. There is a number of classical machine learning methods like support vector machines, neural networks, and decision trees, see e.g. [20, 1, 6], to deal with these problems. However, we simultaneously have the ever more important issue of interpretability of the models and therefore finding how the predictions come to pass. With this information one can e.g. decide to not measure a certain value if it has little influence, in the case the measuring would be expensive, or use it to tune certain variables in order to achieve a desired outcome. While there is new research in the area of interpretability of the classical methods, see e.g. [43, 32], those models do not intrinsically allow for it. The method we apply in this paper does not only provide an alternative to classical machine learning methods, but also comes with a natural way to identify the importance and influence of attributes on the outcome.

In this paper we apply the approximation method presented in [37, 38] and propose ways in which to use them to analyze data and create predictions. The method is based on the idea of the analysis of variance (ANOVA) decomposition, cf. [7, 39, 29, 27, 21, 17], which allows us to approximate high-dimensional functions of a grouped structure when paired with the Grouped Transformations idea in [2]. The ANOVA decomposition uniquely decomposes a function into terms that correspond to variable couplings or variable interactions. The basis is the Lebesgue Hilbert space L_2 for periodic functions or non-periodic functions defined over $[0, 1]$ since here we have a complete orthonormal systems of functions and fast algorithms for multiplication in the Grouped Transformations, see [36, 2].

Therefore, the only a-priori assumptions that we need are that the function we wish to approximate is either periodic or defined over a finite interval where it square-integrable as well as that the function can be well explained by limiting the variable interactions. This means that the significant part of the function is explained by letting only up to a number of variables interact simultaneously, see e.g. [7, 27, 11, 19], relating to the concept of the superposition dimension, see [7, 33]. These assumptions are in general not very restricting and allows for a broad range of functions. Moreover, it has been theorized that most real word applications consist only of low-order interactions relating to sparsity-of-effects, cf. [49], or the Pareto principle.

Since the method gives us importance information on the variable couplings by using global sensitivity indices, cf. [45, 46, 29], and sensitivity analysis, see [42], we can not only interpret this information, but moreover use it to

improve the model by a number of techniques. Using attribute rankings we can remove an unimportant variable entirely and reduce the dimensionality of the problem. It is also possible to find that the representation of the data in the groups is sparse, i.e., some specific variable interactions does not influence the approximation (significantly) and can therefore be discarded. These techniques allow us to build an *active set* of couplings that will in the end represent the model we use for predictions. This simultaneously gives us a control mechanism for the complexity of the model and combat overfitting. This technique is also related to low-dimensional structures and active subspace methods [13, 9, 10] as well as random features [40, 8, 50, 28, 19]. The main difference to random features is that it draws weights or in our language indices/frequencies at random and uses a different optimization problem.

We apply the proposed approach to the Friedman functions, see [30, 4, 5], as an example of how it performs on synthetic data with Gaussian noise and compare our findings to previously obtained benchmark results in the same setting. Moreover, we test the method on real datasets for regression problems from the UCI database [12] and other sources. Each datasets provides a different challenge and we compare our results to previously obtained classical machine learning techniques. We observe very promising results and in many cases outperform previous benchmark experiments.

The paper is organizes as follows. In Section 2 we reiterate on the previously in [37, 38, 2] introduced approximation method and how to use it for regression problems. Here, we discuss the required functional analytic foundations and introduce relevant complete orthonormal systems of functions. The procedure is explained in Section 2.1 and we also propose multiple techniques for model refinement and active set detection in Section 2.2 that includes the computation of an attribute ranking. Section 3 contains numerical experiments with synthetic data namely the Friedman functions. We recreate the setting from benchmark results and test our method under the same conditions. Moreover, in Section 4 we test the method on datasets from [12, 47] and compare results to previous experiments. All numerical experiments are performed with the Julia package [3] and the code is available in the supplementary material as a snapshot as well as online as a repository [44].

2 Interpretable ANOVA Approximation

In this section we summarize on the interpretable ANOVA (Analysis of Variance) approximation method and the associated idea of grouped transformations, see [37, 2]. The approach was considered for periodic functions, but has since been expended to non-periodic functions in [38]. Therefore, we are able to utilize both types of approximations which brings advantages especially for real data.

We consider functions

$$f \in \mathrm{L}_2(\mathrm{D}^d) \coloneqq \left\{ f\colon \mathrm{D}^d \to \mathbb{K}\colon \ \|f\|_{\mathrm{L}_2(\mathrm{D}^d)} \coloneqq \sqrt{\int_{\mathrm{D}^d} |f(\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x}} < \infty \right\}$$

with spatial dimension $d \in \mathbb{N}$ where D is the torus $\mathbb{T}$ for 1-periodic functions and $D = [0, 1]$ if we are non-periodic. We identify the torus with an interval of unit length specifically $\mathbb{T} \cong [-0.5, 0.5)$. The function maps to the real numbers, i.e., $\mathbb{K} = \mathbb{R}$, in the non-periodic case while $\mathbb{K} = \mathbb{C}$ is possible for the periodic case. Furthermore, we have the scalar product

$$\langle f, g \rangle \coloneqq \int_{\mathrm{D}^d} f(\boldsymbol{x})\, \overline{g(\boldsymbol{x})} \,\mathrm{d}\boldsymbol{x}.$$

Now, let $\{\phi_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{Z}^d}$ be a complete orthonormal system in the space $\mathrm{L}_2(\mathrm{D}^d)$ with tensor product structure, i.e., we have a complete orthonormal system $\{\eta_k\}_{k\in\mathbb{Z}}$ in $\mathrm{L}_2(\mathrm{D})$ and $\phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{i=1}^d \eta_{k_i}(x_i)$. Then

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} \mathrm{c}_{\boldsymbol{k}}(f)\, \phi_{\boldsymbol{k}}(\boldsymbol{x}), \quad \mathrm{c}_{\boldsymbol{k}}(f) = \langle f, \phi_{\boldsymbol{k}} \rangle, \tag{1}$$

and through Parseval's identity $\|f\|^2_{\mathrm{L}_2(\mathrm{D}^d)} = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} |\mathrm{c}_{\boldsymbol{k}}(f)|^2$.

The classical ANOVA decomposition, cf. [7, 39, 29, 21], provides us with a unique decomposition in the frequency domain as shown in [37]. We denote the coordinate indices with $\mathcal{D} = \{1, 2, \ldots, d\}$ and subsets as bold small letters, e.g., $\boldsymbol{u} \subseteq \mathcal{D}$. An ANOVA term is defined as

$$f_{\boldsymbol{u}}(\boldsymbol{x}) = f_{\boldsymbol{u}}(\boldsymbol{x}_{\boldsymbol{u}}) \coloneqq \sum_{\substack{\boldsymbol{k}\in\mathbb{Z}^d \\ \operatorname{supp}\boldsymbol{k}=\boldsymbol{u}}} \mathrm{c}_{\boldsymbol{k}}(f)\, \phi_{\boldsymbol{k}}(\boldsymbol{x})$$

with $\operatorname{supp}\boldsymbol{k} \coloneqq \{s \in \mathcal{D}\colon k_s \neq 0\}$ The function can then be uniquely decomposed as

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{u}\subseteq\mathcal{D}} f_{\boldsymbol{u}}(\boldsymbol{x})$$

into $|\mathcal{P}(\mathcal{D})| = 2^d$ ANOVA terms where $\mathcal{P}(\mathcal{D})$ is the potency set of $\mathcal{D}$. Here, the exponentially growing number of terms shows an expression of the curse of dimensionality in the decompositon.

Crucial information to later achieve attribute rankings is the relative importance of an ANOVA terms $f_{\boldsymbol{u}}$ with respect to the function. In order to measure this we define the variance of a function f as

$$\sigma^2(f) \coloneqq \|f\|^2_{\mathrm{L}_2(\mathrm{D}^d)} - |\mathrm{c}_{\mathbf{0}}(f)|^2 = \sum_{\boldsymbol{k}\in\mathbb{Z}^d\setminus\{\mathbf{0}\}} |\mathrm{c}_{\boldsymbol{k}}(f)|^2 .$$

Note that $\sigma^2(f_{\boldsymbol{u}}) = \|f_{\boldsymbol{u}}\|^2_{\mathrm{L}_2(\mathrm{D}^d)}$, $\boldsymbol{u} \subseteq \mathcal{D}$. The relative importance is now measured through global sensitivity indices (GSI), see [45, 46, 29], defined as

$$\varrho(\boldsymbol{u}, f) \coloneqq \frac{\sigma^2(f_{\boldsymbol{u}})}{\sigma^2(f)}. \tag{2}$$

This motivates the concept of effective dimensions. We focus specifically on the modified version of the superposition dimension as one notion of effective dimension. For a given $\alpha \in [0, 1]$ it is defined as

$$\mathrm{d}^{(\mathrm{sp})} \coloneqq \min\left\{ s \in \mathcal{D}\colon \sup_{\|f\|_{H(\mathrm{D}^d)}\leq 1} \sum_{|\boldsymbol{u}|>s} \|f_{\boldsymbol{u}}\|^2_{\mathrm{L}_2(\mathrm{D}^d)} \leq 1-\alpha \right\} \tag{3}$$

for a function $f \in H(\mathrm{D}^d) \subseteq \mathrm{L}_2(\mathrm{D}^d)$. Here, $H(\mathrm{D}^d)$ is $\mathrm{L}_2(\mathrm{D}^d)$ or might be a Hilbert space that e.g. characterizes smoothness trough the decay of the basis coefficients $\mathrm{c}_{\boldsymbol{k}}(f)$, cf. [37, 33].

Now, we still have the curse of dimensionality and need to find a way around it for efficient approximation. For this we take subsets of ANOVA terms $U \subseteq \mathcal{P}(\mathcal{D})$ into account. These sets have to be downward closed, i.e., for every $\boldsymbol{u} \in U$ it holds that all subsets $\boldsymbol{v} \subseteq \boldsymbol{u}$ are also elements of U. We are then able to consider the truncated ANOVA decomposition

$$\mathrm{T}_U f(\boldsymbol{x}) = \sum_{\boldsymbol{u}\in U} f_{\boldsymbol{u}}(\boldsymbol{x}).$$

A specific idea for the truncation comes from the superposition dimension $\mathrm{d}^{(\mathrm{sp})}$ in (3). One might only take variable interactions into account that contain d_s or less variables, i.e., the subset of ANOVA terms is

$$U_{d_s} = \{\boldsymbol{u} \subseteq \mathcal{D}\colon\ |\boldsymbol{u}| \leq d_s\}.$$

Since d_s can be any integer in $\mathcal{D}$ we call it superposition threshold. Note that d_s can be equal to the superposition dimension, but this does not need to be the case. A well-known fact from learning theory is that the number of terms in U_{d_s} grows only polynomially in d for fixed $d_s < d$ which has reduced the curse of dimensionality.

Why is the truncation through a superposition threshold d_s a good idea? Let us start this argument with the approximations of smooth functions that belong to some smoothness space $H(\mathrm{D}^d) \subseteq \mathrm{L}_2(\mathrm{D}^d)$. If the function has a low superposition dimension $\mathrm{d}^{(\mathrm{sp})}$ for $\alpha \in [0, 1]$ the truncation by a low superposition threshold will be effective (in relation to α). It is possible to characterize smoothness by the decay of the basis coefficients $\mathrm{c}_{\boldsymbol{k}}(f)$ and show upper bounds for the superposition dimension $\mathrm{d}^{(\mathrm{sp})}$ as in [37]. In fact there are types of smoothness that are proven to yield a low upper bound for the superposition dimension specifically dominating-mixed smoothness with POD (product and order-dependent) weights, cf. [26, 15, 25, 16, 37].

In terms of real data, the situation is much different. One thing in advance: For the complete generality of problems one cannot make the assumption that we have a low superposition dimension. However, there are many application scenarios where numerical experiments successfully showed that this is indeed the case, see e.g. [7]. Since we generally do not have a-priori information we work with low superposition thresholds d_s for truncation and validate on a set of testdata.

2.1 Approximation Procedure

We briefly discuss how an approximation is numerically obtained in a multiple step procedure and how we can interprete the results as well as create attribute rankings. In this section, we assume a given subset of ANOVA terms $U \subseteq \mathcal{P}(\mathcal{D})$. How to obtain such a set will be discussed in Section 2.2. Moreover, we always assume that we have given scattered data in the form of a node set $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\} \subseteq \mathrm{D}^d$ and values $\boldsymbol{y} \in \mathbb{K}^M$, $M \in \mathbb{N}$. We assume now that there is an $\mathrm{L}_2(\mathrm{D}^d)$ function f of form (1) with $f(\boldsymbol{x}_i) \approx y_i$ which we want to approximate.

The approximation procedure works as follows: We truncate f to the terms in the set U such that $f \approx \mathrm{T}_U f$. Since there are still infinitely many coefficients, we perform a truncation to partial sums with a finite support index set $I_{\boldsymbol{u}} \subseteq (\mathbb{Z} \setminus \{\mathbf{0}\})^{|\boldsymbol{u}|}$ for every ANOVA term $f_{\boldsymbol{u}}$, $\boldsymbol{u} \in U$, such that

$$f_{\boldsymbol{u}}(\boldsymbol{x}) \approx \sum_{\boldsymbol{k} \in P_{\boldsymbol{u}} I_{\boldsymbol{u}}} \mathrm{c}_{\boldsymbol{k}}(f)\, \phi_{\boldsymbol{k}}(\boldsymbol{x})$$

and $P_{\boldsymbol{u}} I_{\boldsymbol{u}} = \{\boldsymbol{k} \in \mathbb{Z}^d \colon \boldsymbol{k}_{\boldsymbol{u}} \in I_{\boldsymbol{u}}, \boldsymbol{k}_{\boldsymbol{u}^{\mathrm{c}}} = \mathbf{0}\}$. Taking the union $I(U) = \bigcup_{\boldsymbol{u} \in U} I_{\boldsymbol{u}}$, we have $f(\boldsymbol{x}) \approx \sum_{\boldsymbol{k} \in I(U)} \mathrm{c}_{\boldsymbol{k}}(f)\, \phi_{\boldsymbol{k}}(\boldsymbol{x})$. The unknown coefficients $\mathrm{c}_{\boldsymbol{k}}(f)$ are now to be determined.

We aim to achieve this by solving the least-squares problem

$$\hat{\boldsymbol{f}} = (\hat{f}_{\boldsymbol{k}})_{\boldsymbol{k} \in I(U)} = \operatorname*{arg\,min}_{\hat{\boldsymbol{g}} \in \mathbb{K}^{|I(U)|}} \|\boldsymbol{y} - \boldsymbol{F}(\mathcal{X}, I(U))\hat{\boldsymbol{g}}\|_2^2 \tag{4}$$

with the matrix $\boldsymbol{F}(\mathcal{X}, I(U)) = (\phi_{\boldsymbol{k}}(\boldsymbol{x}))_{\boldsymbol{k} \in I(U), \boldsymbol{x} \in \mathcal{X}}$. If $\boldsymbol{F}(\mathcal{X}, I(U))$ has full rank the problem has a unique solution and $\hat{f}_{\boldsymbol{k}} \approx \mathrm{c}_{\boldsymbol{k}}(f)$. Then our approximation is

$$f(\boldsymbol{x}) \approx S(\mathcal{X}, I(U)) f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in I(U)} \hat{f}_{\boldsymbol{k}} \phi_{\boldsymbol{k}}(\boldsymbol{x}).$$

In general, if the oversampling $M/\,|I(U)| > 1$, i.e., is large enough and the nodes are independent, we assume full rank for $\boldsymbol{F}(\mathcal{X}, I(U))$. In the case of stochastic matrices where the nodes are drawn i.i.d. according to the density of the space, we refer to [37, Theorem 6.7] for the Fourier system in the periodic case. In [31, Theorem 2.2] which is an application of the Matrix Chernoff, see [48, Theorem 1.1], we have general conditions for the singular values of these matrix being bounded away from zero and therefore having full rank.

How can (4) **be efficiently solved?** In order to solve the minimization we employ the iterative LSQR solver [34] which needs a method for efficient multiplication with $\boldsymbol{F}(\mathcal{X}, I(U))$ and its adjoint $\boldsymbol{F}^*(\mathcal{X}, I(U))$ in the periodic case, otherwise its transposed matrix. This is realized by the Grouped Transformation idea in [2] based on the NFFT or the NFCT, see [22, 36]. Note that we use the matrix-free variant of LSQR, i.e., we never explicitly construct the matrix $\boldsymbol{F}(\mathcal{X}, I(U))$. The grouped transformations provide *oracle* functions for the multiplications.

We consider three distinct function systems in this paper for which we have a fast Grouped Transform. In the periodic case we use the Fourier system with

$$\phi_{\boldsymbol{k}}^{\exp}(\boldsymbol{x}) = \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}. \tag{5}$$

For non-periodic functions we focus on the cosine system

$$\phi_{\boldsymbol{k}}^{\cos}(\boldsymbol{x}) = \sqrt{2}^{|\operatorname{supp} \boldsymbol{k}|} \prod_{s \in \operatorname{supp} \boldsymbol{k}} \cos(\pi k_s x_s) \tag{6}$$

and the Chebyshev system

$$\phi_{\boldsymbol{k}}^{\mathrm{cheb}}(\boldsymbol{x}) = \sqrt{2}^{|\operatorname{supp} \boldsymbol{k}|} \prod_{s \in \operatorname{supp} \boldsymbol{k}} \cos(k_i \arccos(2x_s - 1)). \tag{7}$$

Remark 2.1. *The Chebyshev system was considered in [38]: Although the theoretical approximation properties of this system are better, the nodes $\mathcal{X}$ optimally need to be distributed according to the Chebyshev probability measure and our experiments showed that this did not work well for problems with scattered data from real applications. In [38] we propose a way to circumvent this, by minimizing a weighted norm. However, this has negative effects on the condition of the system and results in the need for more data and/or iterations. Therefore, we opt to use the cosine system for scattered data. However, the situation is quite different if the nodes can be generated according a probability measure, e.g., in uncertainty quantification with PDE applications.*

We also restrict ourselves to the usage of full-grid index sets, i.e., we use

$$I_{\boldsymbol{u}}^{\text{per}} = \{-N_{\boldsymbol{u}}/2, \ldots, -1, 1, \ldots, N_{\boldsymbol{u}}/2 - 1\}$$

in the periodic case and

$$I_{\boldsymbol{u}}^{\text{non-per}} = \{1, 2, \ldots, N_{\boldsymbol{u}} - 1\} \tag{8}$$

for the non-periodic case with the even parameter $N_{\boldsymbol{u}} \in 2\mathbb{N}$. Note that the sets contain $N_{\boldsymbol{u}} - 1$ elements in both cases.

It is possible to add different types of regularization to the problem (4) which allows us to consider systems where full rank of the matrix $\boldsymbol{F}(\mathcal{X}, I(U))$ cannot be guaranteed in particular the underdetermined setting with $|I(U)| > |\mathcal{X}|$.

Moreover, it is possible to incorporate a-priori smoothness information. For details we refer to [2]. For our numerical experiments, we choose to add ℓ_2 or *Tikhonov* regularization, i.e., solving

$$\underset{\hat{\boldsymbol{g}}\in\mathbb{K}^{|I(U)|}}{\arg\min} \|\boldsymbol{y} - \boldsymbol{F}(\mathcal{X}, I(U))\hat{\boldsymbol{g}}\|_2^2 + \lambda \|\hat{\boldsymbol{g}}\|_2^2 \tag{9}$$

with regularization parameter $\lambda > 0$. In order to apply LSQR, we rewrite (9) by observing the equality

$$\|\boldsymbol{y} - \boldsymbol{F}(\mathcal{X}, I(U))\hat{\boldsymbol{g}}\|_2^2 + \lambda \|\hat{\boldsymbol{g}}\|_2^2 = \left\| \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix} - \begin{pmatrix} \boldsymbol{F}(\mathcal{X}, I(U)) \\ \sqrt{\lambda}\boldsymbol{I} \end{pmatrix} \hat{\boldsymbol{g}} \right\|_2^2 \tag{10}$$

with $\boldsymbol{0}$ the zero vector in $\mathbb{R}^{|I(U)|}$ and $\boldsymbol{I} \in \mathbb{R}^{|I(U)|,|I(U)|}$ the identity matrix. Note that we always have a unique solution in this case since the matrix

$$\begin{pmatrix} \boldsymbol{F}(\mathcal{X}, I(U)) \\ \sqrt{\lambda}\boldsymbol{I} \end{pmatrix}$$

always has full column rank.

How can an attribute ranking be computed? We use the global sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}, I(U))f)$, $\boldsymbol{u} \in U$, from the approximation $S(\mathcal{X}, I(U))f(\boldsymbol{x})$ to compute approximations for the global sensitivity indices $\varrho(\boldsymbol{u}, f)$ of the function f. Here, we do not consider the index to be a good approximation if the values are close together, but rather if there order is identical, i.e., we have

$$\varrho(\boldsymbol{u}_1, f) \leq \varrho(\boldsymbol{u}_2, f) \Longrightarrow \varrho(\boldsymbol{u}_1, S(\mathcal{X}, I(U))f) \leq \varrho(\boldsymbol{u}_2, S(\mathcal{X}, I(U))f)$$

for any pair $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U$. We assume that this is the case for our choices of index sets $I(U)$. The accuracy of this assumption will be investigated in Section 3.

If we are interested in how much one variable $i \in \mathcal{D}$ adds to the variance of the function, i.e., how important it is, we sum over all contributing sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}, I(U))f)$ with $i \in \boldsymbol{u}$. This leads to the sum

$$\sum_{\boldsymbol{u}\in\{\boldsymbol{v}\in U\colon\, i\in\boldsymbol{v}\}} \varrho(\boldsymbol{u}, S(\mathcal{X}, I(U))f).$$

In order to obtain a ranking score, we weigh the sensitivity indices by the number of sets $\boldsymbol{v} \in U$ with $|\boldsymbol{u}| = |\boldsymbol{v}|$ and $i \in \boldsymbol{v}$. This counteracts an effect that makes a variable seem important if it occurs in many unimportant terms of the same order in contrast to other variables with some terms of that order not appearing in U. We obtain the ranking score

$$r(i) = \frac{\sum_{\boldsymbol{u}\in\{\boldsymbol{v}\in U\colon\, i\in\boldsymbol{v}\}} |\{\boldsymbol{v}\in U\colon\, |\boldsymbol{u}| = |\boldsymbol{v}|\,, i\in\boldsymbol{v}\}|^{-1}\, \varrho(\boldsymbol{u}, S(\mathcal{X}, I(U))f)}{\sum_{\boldsymbol{u}\in U}\left(\sum_{i\in\boldsymbol{u}} |\{\boldsymbol{v}\in U\colon\, |\boldsymbol{u}| = |\boldsymbol{v}|\,, i\in\boldsymbol{v}\}|^{-1}\right) \varrho(\boldsymbol{u}, S(\mathcal{X}, I(U))f)}. \tag{11}$$

In the denominator we added normalization such that $\sum_{i\in\mathcal{D}} r(i) = 1$. Computing every score $r(i)$, $i \in \mathcal{D}$ provides an attribute ranking with respect to U showing the percentage that every variable adds to the variance of the approximation. This allows for the conclusion that if we have a good approximation $S(\mathcal{X}, I(U))f$, its attribute ranking will be close to the attribute ranking of the function f.

2.2 Active Set Detection

In this section we describe how to obtain an active set of ANOVA terms U for approximation. We are sill working with scattered data $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\} \subseteq \mathrm{D}^d$ and $\boldsymbol{y} \in \mathbb{K}^M$, $M \in \mathbb{N}$. The values $\boldsymbol{y}$ may also contain noise.

Why does it make sense to reduce the number of ANOVA terms?

The first step is to limit the variable interactions by a superposition threshold $d_s \in \mathcal{D}$ which may have been estimated by known smoothness properties (or different a-priori information) or set to a sensible value if nothing is known. Of course it is also possible to test and validate different values. We use the procedure described in Section 2.1 to obtain the approximation $S(\mathcal{X}, I(U_{d_s}))f$. For that we choose index sets $I_{\boldsymbol{u}}$ in an order-dependent way since there is no additional information available, i.e., $N_{\boldsymbol{u}_1} = N_{\boldsymbol{u}_2}$ for $|\boldsymbol{u}_1| = |\boldsymbol{u}_2|$. In order to get more consistence in the global sensitivity indices it is advisable to choose the index sets roughly of the same size, i.e., $(N_{\boldsymbol{u}_1} - 1)^{|\boldsymbol{u}_1|} \approx (N_{\boldsymbol{u}_2} - 1)^{|\boldsymbol{u}_2|}$ for $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U_{d_s}$.

From the approximation $S(\mathcal{X}, I(U_{d_s}))f$ we obtain the global sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}, I(U_{d_s}))f)$, $\boldsymbol{u} \in U$, and an attribute ranking $r(i)$, $i \in \mathcal{D}$, see (11). There are multiple ways to proceed from this point which we explain in the following.

Removal of unimportant variables: If the attribute ranking $r(i)$ shows variables that have very little to no influence on the variance of the function, then those variables may be removed entirely. Removing them leads to a reduction in the dimensionality of the problem and greatly simplifies the model function.

Active Set Thresholding: Here, one chooses a threshold vector $\boldsymbol{\varepsilon} \in (0,1)^{d_s}$ and reduces the ANOVA terms to the set

$$U(\boldsymbol{\varepsilon}) \coloneqq \left\{ \boldsymbol{u} \in U_{d_s} \colon \varrho(\boldsymbol{u}, S(\mathcal{X}, I(U_{d_s}))f) > \varepsilon_{|\boldsymbol{u}|} \right\}.$$

Here, $\varepsilon_{|\boldsymbol{u}|}$ denotes the $|\boldsymbol{u}|$th entry of the vector $\boldsymbol{\varepsilon}$. This set is not downward closed by definition, but if we set for all subsets $\boldsymbol{v}$ of the sets $\boldsymbol{u} \in U(\boldsymbol{\varepsilon})$ with $\boldsymbol{v} \notin U(\boldsymbol{\varepsilon})$ that $f_{\boldsymbol{v}} \equiv 0$ the condition is fulfilled. The parameter vector $\boldsymbol{\varepsilon} \in (0,1)^{d_s}$ allows control over how much of the variance may be sacrificed in order to simplify the model function.

Incremental Expansion: This method is advantageous if the model function is already very complex with a small superposition threshold d_s which may occur if we are dealing with an especially large spatial dimension d. Here, we start with a small d_s, e.g., $d_s = 1$ or $d_s = 2$. A reduction in the ANOVA terms can be performed by either of the two previous approaches. Now, one chooses a $\theta \in (0,1)$ and determines the subset

$$\boldsymbol{v} \coloneqq \{ i \in \mathcal{D} \colon r(i) > \theta \}.$$

If we assume that additional interactions of the important variables might also be significant to the variance, we may add interactions of size up to $n_{\boldsymbol{v}} \in \mathbb{N}$, $d_s < n_{\boldsymbol{v}} < d$. This translates to adding the set of terms

$$U(\boldsymbol{v}, n_{\boldsymbol{v}}) \coloneqq \{ \boldsymbol{u} \in \mathcal{P}(\boldsymbol{v}) \colon d_s < |\boldsymbol{u}| \leq n_{\boldsymbol{v}} \}.$$

This will be a beneficial way to improve the accuracy of the model if higher-order interactions play a role. However, if this method is used without reducing the complexity with any of the previous approaches, the overall complexity of the model will be higher.

In summary, one has to interpret the information obtained from the approximation $S(\mathcal{X}, I(U_{d_s}))f$ and choose the best performing methods based on this. Any combination of the previously mentioned approaches may lead to the optimal approximation as we will see in the numerical experiments. Moreover, it is possible and advisable to iterate this procedure multiple times, i.e., cross-validate it with the proposed active set detection steps in order to obtain the best result.

3 Numerical Experiments with Synthetic Data

In this section we test our approximation and attribute ranking idea on synthetic data, i.e., we have a function $f \colon \mathrm{D}^d \to \mathbb{R}$ and approximate it from artificially generated scattered data. Here, we focus on the non-periodic setting, i.e., $\mathrm{D} = [0,1]$ and $\mathbb{K} = \mathbb{R}$, and the cosine basis $\phi_{\boldsymbol{k}}^{\cos}(\boldsymbol{x})$, see (6). Note that extensive tests for the periodic setting with the Fourier system (5) and the non-periodic Chebyshev system (7) have been conducted in [37, 38, 2]. For a fixed number of nodes $M \in \mathbb{N}$ we create the node set $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\} \subseteq [0,1]^d$ by drawing uniform i.i.d. nodes and evaluate the function $\boldsymbol{y} = (f(\boldsymbol{x}_i) + \eta)_{\mathrm{i}=1}^M$. Moreover, the evaluations $\boldsymbol{y}$ may also contain noise η.

We are using the Friedmann functions, cf. [30, 4, 5] for our experiments. Our goal is to achieve the most accurate approximation which we will compare to the quality of different approaches known from the literature. In order to measure the quality of the approximation we take a second set of uniformly distributed i.i.d. nodes $\mathcal{X}_{\text{test}} \subseteq [0,1]^d$, $|\mathcal{X}_{\text{test}}| \in \mathbb{N}$, and choose the mean square error (MSE) as a measure of quality which is defined as

$$\mathrm{MSE}(f, \tilde{f}) = \frac{1}{|\mathcal{X}_{\text{test}}|} \sum_{\boldsymbol{x} \in \mathcal{X}_{\text{test}}} \left| f(\boldsymbol{x}) - \tilde{f}(\boldsymbol{x}) \right|^2. \tag{12}$$

Here, $\tilde{f}$ is the approximation or model for f whose quality is to be measured.

In our experiments we will make use of regularization when solving problem (4), i.e., solving the modified problem (9) which is described in more detail in [2]. Note that we rely solely on the ℓ_2 variant of the regularization and the parameter will be denoted with $\lambda > 0$. The optimal choice for this parameter is determined using cross-validation as described in [2]. The experiments have been conducted using the Julia package [3] and the code examples can be found in the supplement as a snapshot as well as online in [44].

3.1 Friedmann Functions

The Friedmann functions were used as benchmark examples in [30] and have since become an often used example in the approximation of functions with scattered data, see e.g. [4, 5]. We start by defining the three non-periodic Friedmann functions over $[0,1]^d$.

The first function

$$f_1\colon [0,1]^{10} \to \mathbb{R},\, f_1(\boldsymbol{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$$

has spatial dimension 10. However, only five of the ten variables have any influence on the function which is the most important information we aim to find with our attribute ranking. Additionally, no more than two variables interact simultaneously, i.e., there will not be an error because of the ANOVA trunction with U_{d_s} for a superposition threshold $d_s = 2$. In other words, setting $\alpha = 1$ yields $\mathrm{d}^{(\mathrm{sp})} = 2$ in (3).

The second function

$$f_2\colon [0,1]^4 \to \mathbb{R},\, f_2(\boldsymbol{x}) = \sqrt{s_1^2(x_1) + \left(s_2(x_2)\cdot x_3 - \frac{1}{s_2(x_2)\cdot s_4(x_4)}\right)^2}$$

has spatial dimension 4 and contains the variable scalings $s_1(x_1) = 100x_1$, $s_2(x_2) = 520\pi x_2 + 40\pi$, and $s_4(x_4) = 10x_4 + 1$. The scalings are necessary since we want to stay in the interval $[0, 1]$ with each variable. As for the Friedmann 1 function, there are at most two variables interacting simultaneously, i.e., $\alpha = 1$ yields $\mathrm{d}^{(\mathrm{sp})} = 2$ in (3) again.

The third and last Friedmann function is given by

$$f_3\colon [0,1]^4 \to \mathbb{R},\, f_3(\boldsymbol{x}) = \arctan\left(\frac{s_2(x_2)\cdot x_3 - (s_2(x_2)\cdot s_4(x_4))^{-1}}{s_1(x_1)}\right)$$

again with spatial dimension $d = 4$ and the same scalings s_1, s_2, and s_4 as before. Here, every term is (analytically) nonzero which means that entire function (without error) can only be reconstructed for a superposition threshold $d_s = 4$.

In order to compare our results to the experiments in [30], we choose to replicate the setting exactly: We use randomly generated sets $\mathcal{X}^{(i)} \subseteq [0,1]^d$ for each of the Friedman functions $i = 1, 2, 3$ with uniformly distributed i.i.d. nodes such that $M = \left|\mathcal{X}^{(i)}\right| = 200$ for the model training. The dimensions are $d = 10$ for Friedman 1, and $d = 4$ for Friedman 2 and 3. Moreover, we add the same Gaussian noise η_i, $i = 1, 2, 3$, as in [30] to the function evaluations with a mean of zero and a variance of $\sigma_1 = 1$ for Friedman 1, $\sigma_2 = 125$ for Friedman 2, and $\sigma_3 = 0.1$ for Friedman 3, i.e., $\boldsymbol{y}^{(i)} = (f_i(\boldsymbol{x}) + \eta_i)_{\boldsymbol{x}\in\mathcal{X}^{(i)}}$. In order to validate the accuracy of the model, we use test sets of randomly drawn uniformly distributed i.i.d. nodes $\mathcal{X}_{\mathrm{test}}^{(i)} \subseteq [0,1]^d$ for each of the Friedman functions $i = 1, 2, 3$ with $M_{\mathrm{test}} = \left|\mathcal{X}_{\mathrm{test}}^{(i)}\right| = 1000$. The function values are again evaluations with Gaussian noise such that $\boldsymbol{y}_{\mathrm{test}}^{(i)} = (f_i(\boldsymbol{x}) + \eta_i)_{\boldsymbol{x}\in\mathcal{X}_{\mathrm{test}}^{(i)}}$.

Table 1 contains the benchmark data from [30] with a support vector machine (SVM), a linear model (lm), a neural network (mnet) and a random forest (rForst) as well as the results with our method (ANOVAapprox). In the following sections we discuss the detailed procedure on how to obtain the models for ANOVAapprox. Note that we have used the non-periodic cosine basis (6).

	svm	lm	mnet	rForst	ANOVAapprox
Friedman 1	4.36	7.71	9.21	6.02	**1.43**
Friedman 2 ($\cdot 10^3$)	18.13	36.15	19.61	21.50	**17.21**
Friedman 3 ($\cdot 10^{-3}$)	23.15	45.42	**18.12**	22.21	20.69

Table 1: Mean squared errors (MSE) for different methods when approximating Friedman functions in [30] compared to ANOVAapprox. The value for ANOVAapprox was obtained by training the model on 100 randomly generated training sets and validating them on 100 randomly generated test sets. All values are the medians of the experiment MSEs and the best value for every function is highlighted.

The results show that the ANOVA approximation method is competitive to the other approaches and delivers the best MSE for Friedman 1 and 2 as well as a close second best MSE for Friedman 3. Note that a set with 200 datapoints is rather small and other experiments used significantly more data, but we aimed to stay in the exact setting of [30].

3.1.1 Friedman 1

The first Friedman function f_1 provides a good challenge for attribute ranking since it is a 10-dimensional function with only 5 variables that influence its value. Moreover, we only have as few as 200 nodes available for approximation. The only known information is the node set $\mathcal{X}^{(1)}$ and the noisy evaluations $\boldsymbol{y}^{(1)}$. The noise is Gaussian with zero mean and variance $\sigma_1 = 1$.

We begin by setting the superposition threshold to $d_s = 2$. Moreover, we choose $I_\emptyset = \{0\}$, and the order-dependent sets

$$I_{\boldsymbol{u}} = \{1, \dots, N_{|\boldsymbol{u}|} - 1\}, N_{|\boldsymbol{u}|} \in \mathbb{N}, \tag{13}$$

for $|\boldsymbol{u}| = 1, 2$. In Figure 1 we have computed an attribute ranking, see (11), for f_1 which clearly shows that the variables x_6 to x_{10} are significantly less important than the others and indicate that we may remove them completely, i.e., the active set for approximation changes to

$$U^{(\mathrm{r})} = \{\boldsymbol{u} \subseteq \{1, 2, 3, 4, 5\} \colon |\boldsymbol{u}| \leq 2\}.$$

Note that we have computed multiple attribute rankings and displayed the one where the corresponding approximation $S(\mathcal{X}^{(1)}, I(U_2))f_1$, $|I(U_2)| = 76$, achieved the best MSE on the test set $\mathcal{X}^{(1)}_{\mathrm{test}}$ of 4.99, i.e., the closest model to the original function f_1.

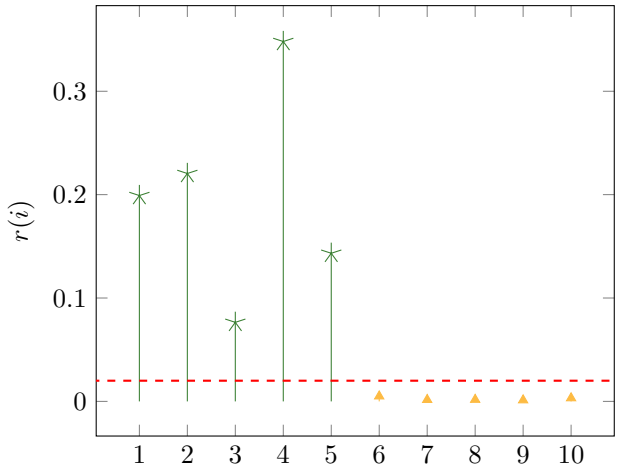


Figure 1: Attribute ranking of the Friedman 1 function using 200 nodes $\mathcal{X}^{(1)}$ and noisy evaluations $\boldsymbol{y}^{(1)}$ with $N_1 = 4$, $N_2 = 2$, regularization paramter $\lambda = 3$, and superposition threshold $d_s = 2$.

We repeat our approximation process with $U^{(\mathrm{r})}$ as active set of terms and consider the approximation $S(\mathcal{X}^{(1)}, I(U^{(\mathrm{r})}))f_1$, still with the parameters N_1, and N_2. Since $1 + \binom{5}{1} + \binom{5}{2} = 16$, we have $\left|U^{(\mathrm{r})}\right| = 16$ and therefore as many global sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}^{(1)}, I(U^{(\mathrm{r})}))f_1)$ to consider. Figure 2 shows the global sensitivity indices for the parameter constellation that yielded the best MSE of 2.50 on the test set $\mathcal{X}^{(1)}_{\mathrm{test}}$. The important sets are

$$U^*_{(1)} = \{\emptyset\} \cup \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2\}\}.$$

while the sets in $U^{(\mathrm{r})} \setminus U^*_{(1)}$ have a smaller global sensitivity index by a large margin and will be removed from the active set in the next step.

We conclude the consideration of the Friedman 1 function with the final approximation $S(\mathcal{X}^{(1)}, I(U^*_{(1)}))f_1$. The results for different N_1, and N_2 are displayed in Table 2. The best MSE on the test set $\mathcal{X}^{(1)}_{\mathrm{test}}$ we were able to obtain is 1.36. As displayed in Table 1, the best MSE achieved by different methods using the same number of nodes and the same noise was 4.36 by a support vector machine. Validating the model with 100 randomly generated training and test datasets yielded a median MSE of 1.43.

In order to justify our decision of using a superposition threshold $d_s = 2$, we repeat the experiments from Figure 1 and Figure 2 with $d_s = 3$. We use $I_\emptyset = \{0\}$, and order-dependent sets $I_{\boldsymbol{u}}$, $|\boldsymbol{u}| = 1, 2, 3$, from (13). In Figure 3a we depicted the attribute ranking where the corresponding approximation $S(\mathcal{X}^{(1)}, I(U_3))f_1$, $|I(U_3)| = 196$, yielded a best MSE of 21.87 on the test set $\mathcal{X}^{(1)}_{\mathrm{test}}$. While the MSE is larger than before, we observe that the attribute ranking clearly shows the importance of variables x_1 to x_5.

We now use the active set

$$\tilde{U}^{(\mathrm{r})} = \{\boldsymbol{u} \subseteq \{1, 2, 3, 4, 5\} \colon |\boldsymbol{u}| \leq 3\}$$

and consider the approximation $S(\mathcal{X}^{(1)}, I(\tilde{U}^{(\mathrm{r})}))f_1$ with parameters N_1, N_2, N_3. The set contains $\left|\tilde{U}^{(\mathrm{r})}\right| = 21$ ANOVA terms. Figure 3b shows the global sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}^{(1)}, I(\tilde{U}^{(\mathrm{r})}))f_1)$ for the parameter choice that yielded

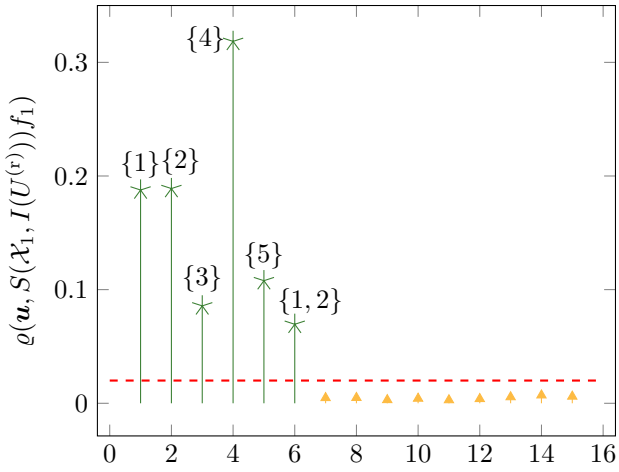


Figure 2: Global sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}^{(1)}, I(U^{(\mathrm{r})}))f_1)$, $\boldsymbol{u} \in U^{(\mathrm{r})}$, with parameters $N_1 = 6$, $N_2 = 4$, and $\lambda = 1$. $U^*_{(1)}$ with circles and complement with rectangles.

N_1	N_2	$\left\lvert I(U^*_{(1)}) \right\rvert$	MSE
4	2	17	3.48
6	2	27	3.50
8	2	37	3.58
4	4	25	1.53
6	4	35	1.36
8	4	45	1.36

Table 2: Numerical experiments with the Friedman 1 function using 200 nodes $\mathcal{X}^{(1)}$ and noisy evaluations $\boldsymbol{y}^{(1)}$ with regularization paramter $\lambda = 1$ and superposition threshold $d_s = 2$. The MSE $\coloneqq \mathrm{MSE}(f_1, S(\mathcal{X}^{(1)}, I(U^*_{(1)}))f_1)$ was computed on the test set $\mathcal{X}^{(1)}_{\mathrm{test}}$ with 1000 nodes.

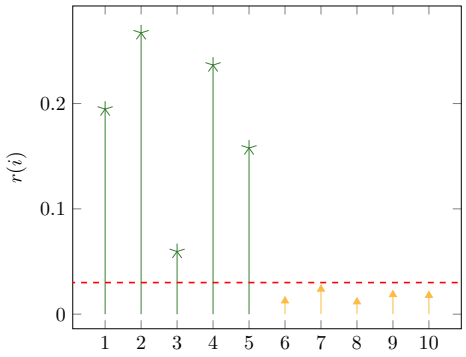
the best MSE 3.04. Note that we have higher MSE than with $d_s = 2$, but we are able to correctly identify the active terms $U^*_{(1)}$. Only a minor part of the variance is attributed to the additional three dimensional variable interactions. In the end, we obtain the same approximation as before.

There exist a number of other methods for attribute rankings or feature selection, see e.g. [18]. We conclude this section with a comparison to the well-known method of the estimation of mutual information. The mutual information between two random variables represents a measure for the dependency between the variables. For a pair of jointly continuous random variables (Z, Y) with values over the space $\mathcal{Z} \times \mathcal{Y}$, it is defined as

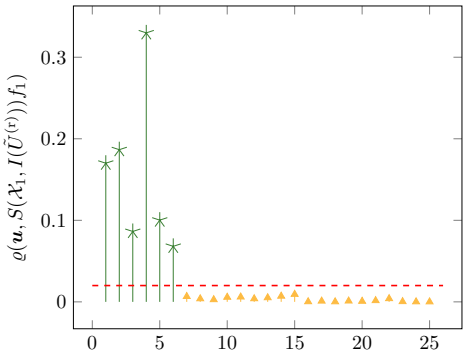
$$I(Z, Y) \coloneqq \int_{\mathcal{Z}} \int_{\mathcal{Y}} p_{(Z,Y)}(z, y) \, \log \frac{p_{(Z,Y)}(z, y)}{p_Z(z) \, p_Y(y)} \, \mathrm{d}z \, \mathrm{d}y$$

with $p_{(Z,Y)}$ the joint probability density function of Z and Y, and p_Z, p_Y the marginal probability density functions of Z and Y, respectively. We apply the function SKLEARN.FEATURE_SELECTION.MUTUAL_INFO_REGRESSION from the scikit-learn Python library, see [35]. The estimation of the mutual information is based on the estimation of entropy from k-nearest neighbor distances, cf. [24, 41].

Table 3 contains the results of our experiments with different values for the number of nearest neighbors k. We have applied the method to the training set $\mathcal{X}^{(1)}$ and the corresponding noisy evaluation $\boldsymbol{y}^{(1)}$. We observe that it is not possible to clearly distinguish the active variables. The active variables x_3 and x_5 get attributed only a small influence while some unimportant variables get attributed a higher influence, e.g., the variable x_10 in the experiment with $k = 5$. In contrast to the experiments with mutual information estimation, our model-based approach by estimating the global sensitivity indices delivers a clear distinction of the active variables on the training set.



(a) Attribute ranking of the Friedman 1 function using 200 nodes $\mathcal{X}^{(1)}$ and noisy evaluations $\boldsymbol{y}^{(1)}$ with $N_1 = 4$, $N_2 = 2$, $N_3 = 2$, regularization paramter $\lambda = 2$, and superposition threshold $d_s = 3$.



(b) Global sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}^{(1)}, I(\tilde{U}^{(\mathrm{r})}))f_1)$, $\boldsymbol{u} \in \tilde{U}^{(\mathrm{r})}$, with parameters $N_1 = 6$, $N_2 = 4$, $N_3 = 2$, and $\lambda = 2$. $U^*_{(1)}$ with stars and complement with triangles.

Figure 3: Repetition of experiments with Friedman 1 function using the higher superposition threshold of $d_s = 3$.

k	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
3	0.14	0.15	0.00	0.21	0.10	0.00	0.00	0.01	0.04	0.05
5	0.17	0.11	0.05	0.23	0.07	0.00	0.00	0.00	0.00	0.11
7	0.14	0.12	0.04	0.25	0.05	0.00	0.03	0.02	0.00	0.04

Table 3: Attribute ranking obtained by mutual information estimation using the scikit-learn library. The parameter k represents the number of neighbors to use in the entropy estimation.

3.1.2 Friedman 2

The second Friedman function f_2 is only four-dimensional with every dimension playing a role in the function. Therefore, we skip the attribute ranking and straightforward try to identify an active set of terms. For that we rely on the 200 nodes $\mathcal{X}^{(2)}$ and the evaluations $\boldsymbol{y}^{(2)}$ with Gaussian noise that has a mean of zero and a variance of $\sigma_2 = 125$.

As for the Friedman 1 function, we set the superposition threshold to $d_s = 2$. We also use full grid index sets $I_\emptyset = \{0\}$, and $I_{\boldsymbol{u}}$ for $|\boldsymbol{u}| = 1, 2$ as in (13) again. In Figure 4 we have visualized the global sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}^{(2)}, I(U_2))f_2)$, $\boldsymbol{u} \in U_2$, for which $S(\mathcal{X}^{(2)}, I(U_2))f_2$ yielded the best MSE of $17.37 \cdot 10^3$ on the test set $\mathcal{X}^{(2)}_{\text{test}}$. We are able to clearly identify the highlighted sets as important and use

$$U^*_{(2)} = \{\emptyset\} \cup \{\{2\}, \{3\}, \{2, 3\}\}.$$

as active set going forward.

We proceed to approximate and show the results for different parameters in Table 4. The best MSE achieved on the test set $\mathcal{X}^{(2)}_{\text{test}}$ is $16.84 \cdot 10^3$ compared to $18.13 \cdot 10^3$ by a support vector machine, see Table 1. Validating the model with 100 randomly generated training and test datasets yielded a median MSE of $17.21 \cdot 10^3$.

3.1.3 Friedman 3

The third Friedman function f_3 provides a challenge since all terms $f_{\boldsymbol{u}}$, $\boldsymbol{u} \subseteq \mathcal{D}$, are nonzero. Therefore, we are making a truncation error if the threshold d_s is smaller than $d = 4$. As before, we use only the 200 nodes in $\mathcal{X}^{(3)}$ and the evaluations $\boldsymbol{y}^{(3)}$ with Gaussian noise that has mean zero and variance $\sigma_3 = 0.1$.

At first we will use a superposition threshold of $d_s = 3$ to identify which ANOVA terms $f_{\boldsymbol{u}}$ with $|\boldsymbol{u}| \leq 3$ are important to the function. As before we rely on full grid index sets $I_\emptyset = \{0\}$, and $I_{\boldsymbol{u}}$ for $|\boldsymbol{u}| = 1, 2, 3$ as in (13). We have visualized the attribute ranking for the parameter choice that yielded the best MSE $2.18 \cdot 10^{-2}$ on the test set $\mathcal{X}^{(3)}_{\text{test}}$ in Figure 5.

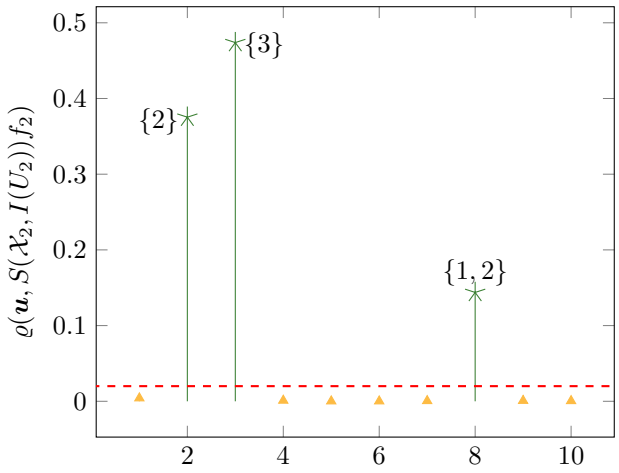


Figure 4: Global sensitivity indices $\varrho(\boldsymbol{u}, S(\mathcal{X}^{(2)}, I(U_2))f_2)$, $\boldsymbol{u} \in U_2$, with parameters $N_1 = 4$, $N_2 = 2$, and $\lambda = 0$. $U^*_{(2)}$ with circles and complement with rectangles.

N_1	N_2	$\left\lvert I(U^*_{(2)}) \right\rvert$	MSE
2	2	4	$18.15 \cdot 10^3$
4	2	8	$16.84 \cdot 10^3$
6	2	12	$16.98 \cdot 10^3$
8	2	16	$17.16 \cdot 10^3$
4	4	16	$17.41 \cdot 10^3$
6	4	20	$17.64 \cdot 10^3$
8	4	24	$17.89 \cdot 10^3$

Table 4: Numerical experiments with the Friedman 2 function using 200 nodes $\mathcal{X}^{(2)}$ and noisy evaluations $\boldsymbol{y}^{(2)}$ with superposition threshold $d_s = 2$. The MSE $\coloneqq \mathrm{MSE}(f_2, S(\mathcal{X}^{(2)}, I(U^*_{(2)}))f_2)$ was computed on the test set $\mathcal{X}^{(2)}_{\mathrm{test}}$ of 1000 nodes.

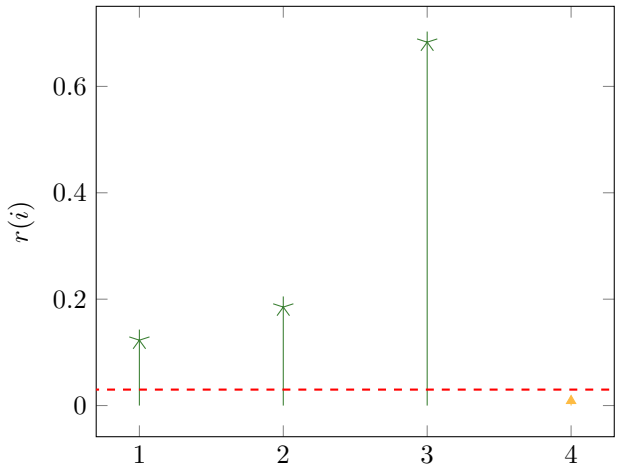


Figure 5: Attribute ranking of the Friedman 3 function using 200 nodes $\mathcal{X}^{(3)}$ and noisy evaluations $\boldsymbol{y}^{(3)}$ with $N_1 = 10$, $N_2 = 2$, $N_3 = 2$, regularization paramter $\lambda = 2$, and superposition threshold $d_s = 3$.

The ranking shows that the variable x_4 has very little influence on the approximation compared to the other 3 variables. This suggests that we are able to use

$$U_{d_s}^{(\mathrm{r})} = \{\boldsymbol{u} \subseteq \{1,2,3\} \colon |\boldsymbol{u}| \leq d_s\}$$

as active set. Table 5 shows the results for approximation of f_3 by $S(\mathcal{X}^{(3)}, I(U_2^{(\mathrm{r})}))f_3$ such that the superposition threshold $d_s = 2$ and $S(\mathcal{X}^{(3)}, I(U_3^{(\mathrm{r})}))f_3$ with $d_s = 3$.

N_1	N_2	N_3	$\left\lvert I(U_2^{(\mathrm{r})})\right\rvert$	$\left\lvert I(U_3^{(\mathrm{r})})\right\rvert$	MSE	$\overline{\mathrm{MSE}}$
10	2	2	31	31	$19.96 \cdot 10^{-3}$	$20.25 \cdot 10^{-3}$
12	2	2	37	38	$19.30 \cdot 10^{-3}$	$19.63 \cdot 10^{-3}$
14	2	2	43	44	$19.68 \cdot 10^{-3}$	$20.06 \cdot 10^{-3}$
10	4	2	55	56	$22.74 \cdot 10^{-3}$	$23.11 \cdot 10^{-3}$
12	4	2	61	62	$22.12 \cdot 10^{-3}$	$22.26 \cdot 10^{-3}$
14	4	2	67	68	$24.26 \cdot 10^{-3}$	$24.33 \cdot 10^{-3}$

Table 5: Numerical experiments with the Friedman 3 function using 200 nodes $\mathcal{X}^{(3)}$ and noisy evaluations $\boldsymbol{y}^{(3)}$. The $\mathrm{MSE} \coloneqq \mathrm{MSE}(f_3, S(\mathcal{X}_3, I(U_2^{(\mathrm{r})}))f_3)$ and $\overline{\mathrm{MSE}} \coloneqq \mathrm{MSE}(f_3, S(\mathcal{X}_3, I(U_3^{(\mathrm{r})}))f_3)$ were computed on the test set $\mathcal{X}_{\mathrm{test}}^{(3)}$ of 1000 nodes.

We achieve a best MSE of $19.3 \cdot 10^{-3}$ on the test set $\mathcal{X}_{\mathrm{test}}^{(3)}$ with active set $U_2^{(\mathrm{r})}$, i.e., we set the superposition threshold to $d_s = 2$ and do not need the three-dimensional term $\{1,2,3\}$. For comparison we find an MSE of $18.12 \cdot 10^{-3}$ as the best result in [30], cf. Table 1. Validating the model with 100 randomly generated training and test datasets yielded a median MSE of $18.12 \cdot 10^{-3}$.

3.2 Runtime

We conclude this section with a runtime comparison of our methods for the experiments conducted with the Friedman functions. Here, we distinguish between an *init* step, i.e., the initialization and pre-computations of the grouped transform, and solving the least-squares problem (9) with LSQR. Table 6 shows the runtimes for computing the approximations. Note that this is the computation for one specific choice of parameters λ, N_1, N_2, and in some cases N_3. However, solving for additional regularization parameters λ does not require a repetition of the initialization. The experiments were conducted on a computer with an Intel Xeon Gold 6240 R CPU at 2.4 gigahertz. We used 12 cores for the parallelization of the grouped transformations.

We observe a direct correlation from the number of involved ANOVA terms to the time for both the initialization and computation step. The experiment with the Friedman 1 function using as superposition threshold $d_s = 3$ consists of 176 ANOVA terms in total with 120 3-dimensional terms. It represents the most expensive experiment with 56.1 seconds for the solution of the least-squares problem in contrast to 2.1 seconds for the $d_s = 2$ experiment with Friedman 1 which uses only 56 terms. As a general observation we notice an increase in computation time with a higher superposition threshold.

4 Numerical Experiments with Real Data

In this section we test the ANOVAapprox method on datasets from real applications, i.e., we get a set of nodes $\mathcal{X} \subseteq [0,1]^d$ with $|\mathcal{X}| = M \in \mathbb{N}$ and noisy evaluations $\boldsymbol{y} \in \mathbb{R}^M$. Note that the data is not in $[0,1]$ in general, but we can achieve this through min-max-normalization in a pre-processing step. Moreover, we have to decide how to split $\mathcal{X}$ into two parts, $\mathcal{X}_{\mathrm{train}}$ which we use for solving the optimization problem, i.e., training our model and obtaining the basis coefficients, and $\mathcal{X}_{\mathrm{test}}$ for validating our method and computing the error. Moreover, we focus on using the non-periodic cosine basis (6) and corresponding index sets $I_{\boldsymbol{u}}^{\text{non-per}}$, see (8), for the ANOVA terms $f_{\boldsymbol{u}}$. Here, $N_{\boldsymbol{u}} \in 2\mathbb{N}$ is the associated bandwidth parameter that we always choose order-dependent, i.e., $N_{\boldsymbol{u}_1} = N_{\boldsymbol{u}_2} = N_{|\boldsymbol{u}_1|}$ for $|\boldsymbol{u}_1| = |\boldsymbol{u}_2|$. As for the synthetic data, we make use of the ℓ_2 regularization proposed in [2], cf. (9). This ensures that the appearing optimization problems have a unique solution since the system matrices have full rank by (10). Note that we used a superposition threshold of $d_s = 2$ for every dataset since we did not observe any improvement in model accuracy when choosing a larger threshold. The experiments have been conducted using the Julia package [3] and the code examples can be found in the supplement as a snapshot as well as online in [44].

function	index set	$\boldsymbol{N}$	init time	LSQR time
Friedman 1	$I(U_2)$	$[4, 2]$	31.6 ms	2.1 s
Friedman 1	$I(U_3)$	$[4, 2, 2]$	81.8 ms	56.1 s
Friedman 1	$I(U^{(\mathrm{r})})$	$[6, 4]$	6.2 ms	0.89 s
Friedman 1	$I(\tilde{U}^{(\mathrm{r})})$	$[6, 4]$	10.4 ms	4.6 s
Friedman 1	$I(U^*_{(1)})$	$[6, 4]$	1.4 ms	54.0 ms
Friedman 2	$I(U_2)$	$[4, 2]$	2.7 ms	85.3 ms
Friedman 2	$I(U^*_{(2)})$	$[4, 2]$	1.3 ms	15.1 ms
Friedman 3	$I(U_3)$	$[10, 2, 2]$	6.2 ms	1.3 s
Friedman 3	$I(U_2^{(\mathrm{r})})$	$[12, 2]$	1.8 ms	61.1 ms
Friedman 3	$I(U_3^{(\mathrm{r})})$	$[12, 2, 2]$	1.6 ms	0.36s

Table 6: Runtimes of experiments with the Friedman functions using the ANOVAapprox method. The pre-computations and initialization is measured as *init time* and computing the solution of the corresponding least-squares problem as *LSQR time* ($\boldsymbol{N} = [N_1, N_2]$ or $\boldsymbol{N} = [N_1, N_2, N_3]$).

Table 7 provides an overview of the data we use and where we obtained it. The datasets are all well-known and have been used for regression benchmarking in the past. We will not provide an in-depth description of the precise approximation steps for every dataset, but rather give a summary and compare the results in the end. Note that we use the root mean square error as a quality measure which describes the square root of the MSE (12), i.e.,

$$\mathrm{RMSE}(f, \tilde{f}) \coloneqq \sqrt{\mathrm{MSE}(f, \tilde{f})}.$$

Moreover, for the *Airfoil Self-Noise* problem we use the relative error

$$\sqrt{\frac{\sum_{\boldsymbol{x} \in \mathcal{X}_{\text{test}}} \left| f(\boldsymbol{x}) - \tilde{f}(\boldsymbol{x}) \right|^2}{\sum_{\boldsymbol{x} \in \mathcal{X}_{\text{test}}} |f(\boldsymbol{x})|^2}}$$

in order to compare our results to [19].

Name	dimension	datapoints	references
ENC	8	768	[14, 12]
ENH	8	768	[14, 12]
Airfoil Self-Noise (ASN)	5	1503	[19, 12]
California Housing (CH)	8	20640	[23, 47]
Ailerons	40	13750	[23, 47]

Table 7: Real datasets for benchmarking the ANOVAapprox method with sources.

Table 8 shows the results we obtained with the ANOVA approximation approach compared to other methods. For the energy efficiency problems *ENC* and *ENH* we compare our results to [14] where different classical machine learning and ensemble methods were tested on the same data. Our results outperform even the ensemble methods when comparing the RMSE. The problem of *Airfoil Self-Noise* was considered as an example in [19] for the newly proposed method of sparse random features. Our obtained model was able to achieve a slightly better result by roughly one percent, see Table 8. The remaining problems *California Housing* and *Ailerons* were considered as benchmark examples for multithreaded local learning regularization networks in [14] where we are also able to achieve a lower RMSE. Note that we have tried to replicate the setting for every dataset, i.e., using the same percentages for training and testing as well as validating our model on 100 random splits. Moreover, Figure 6 contains the attribute rankings for each of the 5 models showing the importance of the different attributes for the datasets.

4.1 Energy Efficiency

The dataset describes the energy efficiency of houses by 8 attributes and two values to predict, the cooling load and the heating load. Therefore, we have two problems ENC (8 attributes, 1 continuous value to predict) for the cooling load

dataset	error (type)	method (reference)	ANOVAapprox
ENC	1.79 (RMSE)	Gradient Boosting Machine ([14])	1.49
ENH	0.48 (RMSE)	Random Forest ([14])	0.44
ASN	0.0277 (relative)	Sparse Random Features ([19])	0.0161
CH	0.11450 (RMSE)	Local Learning Reg. NN ([23])	0.10899
Ailerons	0.04601 (RMSE)	Local Learning Reg. NN ([23])	0.04569

Table 8: Result comparison for different datasets and approaches. The models for ANOVAapprox where validated using 100 random splits of training and test set. More details are discussed in the corresponding subsection of Section 4. The ANOVAapprox error is compared to the best error found in the mentioned source together with the method used therein.

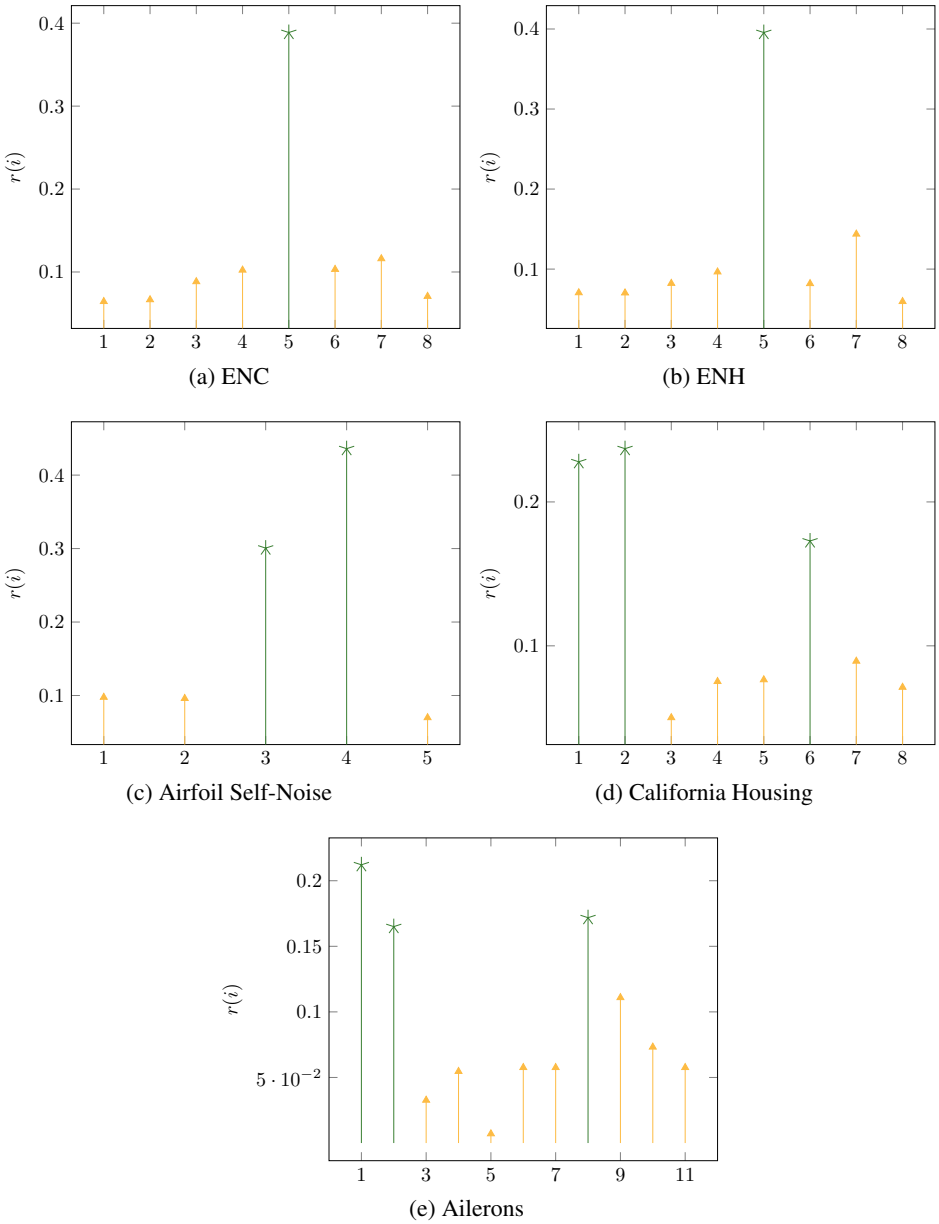


Figure 6: Attribute ranking for the datasets from Table 7.

and ENH (8 attributes, 1 continuous value to predict) for the heating load. The dataset contains 768 samples which we split 70% for the training set $\mathcal{X}_{\text{train}}$ and 30% for the test set $\mathcal{X}_{\text{test}}$. The nodes $\mathcal{X}$ are normalized into $[0, 1]$.

First, we consider the ENC problem and start by setting the superposition threshold $d_s = 2$ and analyze the global sensitivity indices in order to remove unimportant sets. Experiments showed that removing sets or terms with a global sensitivity index (GSI) of less than 0.002 yielded the best result. This leads to an active set $U^*_{\text{ENC}} \subseteq \mathcal{P}(\mathcal{D})$ with 22 terms. The resulting model gives a median RMSE of 1.49 for 100 random splits into training and test set.

For the ENH problem we proceed in a similar fashion. We set the superposition threshold to $d_s = 2$ and then analyze the GSI of our model. Here, we choose the active set $U^*_{\text{ENH}} \subseteq \mathcal{P}(\mathcal{D})$ consisting of all ANOVA terms with a GSI larger than 0.001 such that $|U^*_{\text{ENH}}| = 28$. As a result we obtain a model with a median RMSE of 0.44 for 100 random splits into training and test set.

The optimal order-dependent bandwidths parameters $N_1, N_2 \in \mathbb{N}$ for both problems were computed using cross-validation. Figure 6a and Figure 6b show attribute rankings for our obtained models. We notice that the attribute 5, i.e., the overall height of the building, is especially important for the prediction in both problems.

4.2 Airfoil Self-Noise

This datasets originates from the NASA and contains data about NACA airfoils for different wind tunnel speeds and angles of attack. We aim to find a model that is able to predict the scaled sound pressure level of the self-noise in decibels (continuous), see [12]. The data contains 5 attributes and 1503 nodes. We perform a random split with 80% for the training set $\mathcal{X}_{\text{train}}$ and 20% for the test set $\mathcal{X}_{\text{test}}$. Since this dataset has recently been used in [19] for experiments with sparse random features, we choose the same split to compare the results. Note that the nodes were normalized into $[0, 1]$.

An analysis of the global sensitivity indices for the superposition threshold $d_s = 2$ shows that there is only one unimportant term with a GSI less than 0.001 that is to be removed. Therefore, we have an active set $U^*_{\text{ASN}} \subseteq \mathcal{P}(\mathcal{D})$ with $|U^*_{\text{ASN}}| = 14$ and need to use cross-validation in order to determine the optimal order-dependent bandwidths parameters $N_1, N_2 \in \mathbb{N}$. The obtained model was validated on 100 random 80/20 splits into training and test data yielding a median relative error of 1.61%. In Figure 6c we have visualized the attribute ranking for our model. It shows that attributes 3 and 4, i.e., the chord length and the free-stream velocity have a large influence on the predictions.

4.3 California Housing

The datasets describes the prices for houses in California using data about the block groups from the 1990 census. Using 8 attributes and a set of 20460 cases, we aim to predict the median house price for the area. Since we want to compare our results to [23], we have split the data in 50% for the training set $\mathcal{X}_{\text{train}}$ and another 50% for the test set $\mathcal{X}_{\text{test}}$. The nodes **as well as the evaluations** were normalized into $[0, 1]$. The normalization of the evaluations is replicated from [23].

We used a superposition threshold of $d_s = 2$ and analyzed the GSIs of the ANOVA terms. This lead to an active set $U^*_{\text{CH}} \subseteq \mathcal{P}(\mathcal{D})$ with $|U^*_{\text{CH}}| = 21$ terms. The bandwidth parameters $N_1, N_2 \in \mathbb{N}$ were then computed using cross-validation. The model was subsequently validated on 100 random 50/50 splits of the training and test data which yielded a median RMSE of 0.10899. Figure 6d shows the attribute ranking for the obtained model hinting that the variables 1, 2, and 6, i.e., the geographical location and the population count, are most important for the prediction. It also evident from the GSI that the ANOVA term $f_{\{1,2\}}$ has significant importance which makes sense since variable 1 is the longitude and variable 2 the latitude and together they represent the geographical location.

4.4 Ailerons

The Ailerons dataset describes the control problem of flying a F16 aircraft. The attributes describe the status of the aircraft while we aim to predict the control action on its ailerons. We have 40 attributes and 13750 samples. In order to replicate the setting in [23], we have split the data in 50% for the training set $\mathcal{X}_{\text{train}}$ and another 50% for the test set $\mathcal{X}_{\text{test}}$. The nodes **as well as the evaluations** were normalized into $[0, 1]$. The normalization of the evaluations is replicated from [23].

We started to consider an attribute ranking for superposition threshold $d_s = 1$ in order to check if some variables have little influence and can be omitted for the model. This lead us to eliminate 29 variables with a small contribution. We determined this number through cross-validation. Afterwards, we proceeded with the 11 active variables and $d_s = 2$. A sensitivity analysis leads to the elimination of more terms leading to an active set $U^*_{\text{Ail}} \subseteq \mathcal{P}(\mathcal{D})$ with $|U^*_{\text{Ail}}| = 43$ terms. A validation of our model on 100 random 50/50 splits into training and test data has yielded a median RMSE

of 0.04569. In Figure 6e we have visualized the attribute ranking for our model showing that variables 1,2, and 8 are important. They correspond to the variables 7, 3, and 30 of the original problem.

5 Conclusion

Numerical experiments with synthetic and real data showed that the proposed approach for approximation using ANOVA and Grouped Transformations, see [37, 38, 2], is a competitive method in the approximation of high-dimensional data outperforming even ensemble machine learning methods in our experiments. Moreover, it delivers additional evidence for the fact that in applications we are able to assume that functions consist of (mostly) low-order interactions or are at least explained well by them. Since the method allows intrinsically for interpretation, we are able to produce an attribute ranking that shows how much different attributes influence the predictions. This can also be used to improve the model by removing unimportant variables or variable interactions entirely. Finally, we have proposed and applied multiple methods for the detection of an active set of ANOVA terms.

Acknowledgments

First of all, we thank the referees for their valuable comments and suggestions. The authors also thank their colleagues in the research group SAlE for valuable discussions on the contents of this paper. Daniel Potts acknowledges funding by Deutsche Forschungsgemeinschaft (German Research Foundation) – Project–ID 416228727 – SFB 1410. Michael Schmischke is supported by the BMBF grant 01|S20053A.

References

[1] C. C. AGGARWAL, *Data Classification: Algorithms and Applications*, Chapman & Hall/CRC, 1st ed., 2014.

[2] F. BARTEL, D. POTTS, AND M. SCHMISCHKE, *Grouped transformations in high-dimensional explainable ANOVA approximation*, ArXiv e-prints 2010.10199, (2020).

[3] F. BARTEL AND M. SCHMISCHKE, *ANOVAapprox Julia package*. `https://github.com/NFFT/ANOVAapprox/`, 2020.

[4] G. BEYLKIN, J. GARCKE, AND M. MOHLENKAMP, *Multivariate regression and machine learning with sums of separable functions*, SIAM J. Scientific Computing, 31 (2009), pp. 1840–1857, `https://doi.org/10.1137/070710524`.

[5] P. BINEV, W. DAHMEN, AND P. LAMBY, *Fast high-dimensional approximation with sparse occupancy trees*, J. Comput. Appl. Math., 235 (2011), pp. 2063 – 2076, `https://doi.org/10.1016/j.cam.2010.10.005`.

[6] C. M. BISHOP, *Pattern Recognition and Machine Learning*, Springer New York, Berlin-Heidelberg, 2016.

[7] R. CAFLISCH, W. MOROKOFF, AND A. OWEN, *Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension*, J. Comput. Finance, 1 (1997), pp. 27–46, `https://doi.org/10.21314/jcf.1997.005`.

[8] R. CHITTA, R. JIN, AND A. K. JAIN, *Efficient kernel clustering using random Fourier features*, in 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, `https://doi.org/10.1109/icdm.2012.61`.

[9] P. G. CONSTANTINE, E. DOW, AND Q. WANG, *Active subspace methods in theory and practice: Applications to kriging surfaces*, SIAM J. Sci. Comput., 36 (2014), pp. A1500–A1524, `https://doi.org/10.1137/130916138`.

[10] P. G. CONSTANTINE, A. EFTEKHARI, J. HOKANSON, AND R. A. WARD, *A near-stationary subspace for ridge approximation*, Comput. Methods Appl. Mech. Engrg., 326 (2017), pp. 402–421, `https://doi.org/10.1016/j.cma.2017.07.038`.

[11] R. DEVORE, G. PETROVA, AND P. WOJTASZCZYK, *Approximation of functions of few variables in high dimensions*, Constr. Approx., 33 (2010), pp. 125–143, `https://doi.org/10.1007/s00365-010-9105-8`.

[12] D. DUA AND C. GRAFF, *UCI machine learning repository*, 2017, `http://archive.ics.uci.edu/ml`.

[13] M. FORNASIER, K. SCHNASS, AND J. VYBIRAL, *Learning functions of few arbitrary linear parameters in high dimensions*, Found. Comput. Math., 12 (2012), pp. 229–262, `https://doi.org/10.1007/s10208-012-9115-y`.

[14] M. GOYAL, M. PANDEY, AND R. THAKUR, *Exploratory analysis of machine learning techniques to predict energy efficiency in buildings*, in 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), IEEE, 2020, https://doi.org/10.1109/icrito48877.2020.9197976.

[15] I. G. GRAHAM, F. Y. KUO, J. A. NICHOLS, R. SCHEICHL, C. SCHWAB, AND I. H. SLOAN, *Quasi-Monte Carlo finite element methods for elliptic PDEs with lognormal random coefficients*, Numer. Math., 131 (2014), pp. 329–368, https://doi.org/10.1007/s00211-014-0689-y.

[16] I. G. GRAHAM, F. Y. KUO, D. NUYENS, R. SCHEICHL, AND I. H. SLOAN, *Circulant embedding with QMC: analysis for elliptic PDE with lognormal coefficients*, Numer. Math., 140 (2018), pp. 479–511, https://doi.org/10.1007/s00211-018-0968-0.

[17] C. GU, *Smoothing Spline ANOVA Models*, Springer New York, 2013, https://doi.org/10.1007/978-1-4614-5369-7.

[18] I. GUYON AND A. ELISSEEFF, *An introduction to variable and feature selection*, J. Mach. Learn. Res., 3 (2003), p. 1157–1182.

[19] A. HASHEMI, H. SCHAEFFER, R. SHI, U. TOPCU, G. TRAN, AND R. WARD, *Generalization bounds for sparse random feature expansions*, ArXiv e-prints 2103.03191, (2021).

[20] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*, Springer Science & Business Media, Berlin Heidelberg, 2013.

[21] M. HOLTZ, *Sparse grid quadrature in high dimensions with applications in finance and insurance*, vol. 77 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2011, https://doi.org/10.1007/978-3-642-16004-2.

[22] J. KEINER, S. KUNIS, AND D. POTTS, *Using NFFT3 - a software library for various nonequispaced fast Fourier transforms*, ACM Trans. Math. Software, 36 (2009), pp. Article 19, 1–30, https://doi.org/10.1145/1555386.1555388.

[23] Y. KOKKINOS AND K. G. MARGARITIS, *Multithreaded local learning regularization neural networks for regression tasks*, in Engineering Applications of Neural Networks, Springer International Publishing, 2015, pp. 129–138, https://doi.org/10.1007/978-3-319-23983-5_13.

[24] A. KRASKOV, H. STÖGBAUER, AND P. GRASSBERGER, *Estimating mutual information*, Phys. Rev. E, 69 (2004), https://doi.org/10.1103/physreve.69.066138.

[25] F. Y. KUO AND D. NUYENS, *Application of Quasi-Monte Carlo Methods to Elliptic PDEs with Random Diffusion Coefficients: A Survey of Analysis and Implementation*, Found. Comput. Math, 16 (2016), pp. 1631–1696, https://doi.org/10.1007/s10208-016-9329-5, https://doi.org/10.1007/s10208-016-9329-5.

[26] F. Y. KUO, C. SCHWAB, AND I. H. SLOAN, *Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients*, SIAM J. Numer. Anal., 50 (2012), pp. 3351 – 3374, https://doi.org/10.1137/110845537.

[27] F. Y. KUO, I. H. SLOAN, G. W. WASILKOWSKI, AND H. WOŹNIAKOWSKI, *On decompositions of multivariate functions*, Math. Comp., 79 (2009), pp. 953–966, https://doi.org/10.1090/s0025-5718-09-02319-9.

[28] Z. LI, J.-F. TON, D. OGLIC, AND D. SEJDINOVIC, *Towards a unified analysis of random Fourier features*, in Proceedings of the 36th International Conference on Machine Learning, K. Chaudhuri and R. Salakhutdinov, eds., vol. 97 of Proceedings of Machine Learning Research, PMLR, 09–15 Jun 2019, pp. 3905–3914, http://proceedings.mlr.press/v97/li19k.html.

[29] R. LIU AND A. B. OWEN, *Estimating mean dimensionality of analysis of variance decompositions*, J. Amer. Statist. Assoc., 101 (2006), pp. 712–721, https://doi.org/10.1198/016214505000001410.

[30] D. MEYER, F. LEISCH, AND K. HORNIK, *The support vector machine under test*, Neurocomputing, 55 (2003), pp. 169 – 186, https://doi.org/10.1016/S0925-2312(03)00431-4. Support Vector Machines.

[31] M. MOELLER AND T. ULLRICH, L_2*-norm sampling discretization and recovery of functions from RKHS with finite trace*, Sampl. Theory Signal Process. Data Anal., 19 (2021), https://doi.org/10.1007/s43670-021-00013-3.

[32] G. MONTAVON, W. SAMEK, AND K.-R. MÜLLER, *Methods for interpreting and understanding deep neural networks*, Digit. Signal Process., 73 (2018), pp. 1–15, https://doi.org/10.1016/j.dsp.2017.10.011.

[33] A. OWEN, *Effective dimension of some weighted pre-Sobolev spaces with dominating mixed partial derivatives*, SIAM J. Numer. Anal., 57 (2019), pp. 547–562, https://doi.org/10.1137/17m1158975.

[34] C. C. PAIGE AND M. A. SAUNDERS, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71, https://doi.org/10.1145/355984.355989.

[35] F. PEDREGOSA, G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT, AND E. DUCHESNAY, *Scikit-learn: Machine learning in Python*, J. Mach. Learn. Res., 12 (2011), pp. 2825–2830.

[36] G. PLONKA, D. POTTS, G. STEIDL, AND M. TASCHE, *Numerical Fourier Analysis*, Applied and Numerical Harmonic Analysis, Birkhäuser, 2018, https://doi.org/10.1007/978-3-030-04306-3.

[37] D. POTTS AND M. SCHMISCHKE, *Approximation of high-dimensional periodic functions with Fourier-based methods*, SIAM J. Numer. Anal. (accepted), https://arxiv.org/abs/1907.11412.

[38] D. POTTS AND M. SCHMISCHKE, *Learning multivariate functions with low-dimensional structures using polynomial bases*, ArXiv e-prints 1912.03195, (2019).

[39] H. RABITZ AND O. F. ALIS, *General foundations of high dimensional model representations*, J. Math. Chem., 25 (1999), pp. 197–233, https://doi.org/10.1023/A:1019188517934.

[40] A. RAHIMI AND B. RECHT, *Random features for large-scale kernel machines*, in Advances in Neural Information Processing Systems, J. Platt, D. Koller, Y. Singer, and S. Roweis, eds., vol. 20, Curran Associates, Inc., 2008, https://proceedings.neurips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

[41] B. C. ROSS, *Mutual information between discrete and continuous data sets*, PLoS ONE, 9 (2014), p. e87357, https://doi.org/10.1371/journal.pone.0087357.

[42] A. SALTELLI, M. RATTO, T. ANDRES, F. CAMPOLONGO, J. CARIBONI, D. GATELLI, M. SAISANA, AND S. TARANTOLA, *Global sensitivity analysis: the primer*, John Wiley & Sons, Ltd., 2008.

[43] W. SAMEK, T. WIEGAND, AND K.-R. MÜLLER, *Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models*, ArXiv e-prints 1708.08296, (2017).

[44] M. SCHMISCHKE, *ANOVAapprox numerical experiments*. https://github.com/NFFT/AttributeRankingExamples, 2021.

[45] I. M. SOBOL, *On sensitivity estimation for nonlinear mathematical models*, Keldysh AppliedMathematics Institute, 1 (1990), pp. 112–118.

[46] I. M. SOBOL, *Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates*, Math. Comput. Simulation, 55 (2001), pp. 271–280, https://doi.org/10.1016/s0378-4754(00)00270-6.

[47] L. TORGO, *Regression datasets*. https://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html.

[48] J. A. TROPP, *User-friendly tail bounds for sums of random matrices*, Found. Comput. Math., 12 (2011), pp. 389–434, https://doi.org/10.1007/s10208-011-9099-z.

[49] C. F. J. WU AND M. S. HAMADA, *Experiments - Planning, Analysis, and Optimization*, John Wiley & Sons, New York, 2011.

[50] 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 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, Red Hook, NY, USA, 2012, Curran Associates Inc., p. 476–484.