Sampling Theory, Signal Processing, and Data Analysis (2023) 21:27
https://doi.org/10.1007/s43670-023-00063-9

ORIGINAL PAPER

HARFE: hard-ridge random feature expansion

Esha Saha[1] · **Hayden Schaeffer**[2] · **Giang Tran**[1]

Received: 29 September 2022 / Accepted: 5 July 2023 / Published online: 9 August 2023

Abstract
We propose a random feature model for approximating high-dimensional sparse additive functions called the hard-ridge random feature expansion method (HARFE). This method utilizes a hard-thresholding pursuit-based algorithm applied to the sparse ridge regression (SRR) problem to approximate the coefficients with respect to the random feature matrix. The SRR formulation balances between obtaining sparse models that use fewer terms in their representation and ridge-based smoothing that tend to be robust to noise and outliers. In addition, we use a random sparse connectivity pattern in the random feature matrix to match the additive function assumption. We prove that the HARFE method is guaranteed to converge with a given error bound depending on the noise and the parameters of the sparse ridge regression model. In addition, we provide a risk bound on the learned model. Based on numerical results on synthetic data as well as on real datasets, the HARFE approach obtains lower (or comparable) error than other state-of-the-art algorithms.

Keywords Sparse additive modeling · Random feature expansion · Ridge regression

Mathematics Subject Classification 65D15 · 65D40 · 65K10

1 Introduction

Kernel-based approaches have been extensively used in data-based applications, including image classification and high-dimensional function approximations since they often perform well in practice. These approaches utilize a pre-defined nonlinear function basis in the form of a kernel $K(\mathbf{x}, \mathbf{y})$. Using the representer theorem, minimizers of kernel training problems over reproducing kernel Hilbert spaces (RKHS)

Communicated by Wenjing Liao.

✉ Hayden Schaeffer
hayden@math.ucla.edu

[1] Department of Applied Mathematics, University of Waterloo, Waterloo, Canada

[2] Department of Mathematics, University of California, Los Angeles, USA

take the form of linear combinations of the kernel basis applied to the dataset [11, 24, 43, 50]. Since this technique requires one to apply the kernel K to every pair of data samples, the methods scale quadratically with the size of the dataset, which can limit their use in large-scale applications. A more tractable approach utilizes low-dimensional approximations (or factorizations) of the kernel matrix through the use of randomized features.

The random feature model (RFM) [38–40] is a popular technique for approximating the kernel (and thus the minimizer of kernel regression problems) using a randomized basis that can avoid the cost of full kernel methods. If the kernel is positive definite, then it can be written as an inner product with a feature function ϕ, i.e. $K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$. With RFM, a randomized feature map ψ is used to approximate the kernel $K(\mathbf{x}, \mathbf{y}) \approx \psi(\mathbf{x})^T \psi(\mathbf{y})$, where $\psi : \mathbb{R}^d \to \mathbb{R}^N$. When the random feature map is low-dimensional, i.e. when N is not large, then the approximation is tractable for large-scale datasets. This is the method used for random feature kernel regression.

An alternative perspective is to view the RFM as a nonlinear randomized function approximation. From the neural network point of view, an RFM is a two-layer network with a randomized but fixed single hidden layer [38–40]. Given an unknown function $f : \mathbb{R}^d \to \mathbb{C}$, the RFM takes the form $y = \mathbf{c}^T \phi(\mathbf{W}^T \mathbf{x}) \in \mathbb{C}$ where $\mathbf{x} \in \mathbb{R}^d$ is the input data, $\mathbf{W} \in \mathbb{R}^{d \times N}$ is a random weight matrix, ϕ is a non-linear function applied element-wise and $\mathbf{c} \in \mathbb{C}^N$ is the final weight layer. We take as an assumption that the entries of the matrix $\mathbf{W} = [\boldsymbol{\omega}_{j,k}]$ are independent and identically distributed (i.i.d.) random variables generated by the (user defined) probability density function $\rho(\boldsymbol{\omega})$ i.e. $\boldsymbol{\omega}_{j,k} \sim \rho(\boldsymbol{\omega})$ for all $1 \leq j \leq d$ and $1 \leq k \leq N$. For an RFM, the output layer $\mathbf{c}$ is trained, while the hidden layer $\mathbf{W}$ is fixed. Thus, given a collection of m measurements, whose inputs (and outputs) are arranged column-wise in the matrix $\mathbf{X} \in \mathbb{R}^{d \times m}$ (and $\mathbf{Y} \in \mathbb{C}^{1 \times m}$), the random feature regression problem becomes training $\mathbf{c}$ by optimizing:

$$\min_{\mathbf{c} \in \mathbb{C}^N} \|\mathbf{Y} - \mathbf{c}^T \phi(\mathbf{W}^T \mathbf{X})\|_{\ell^2}^2 + \mathcal{R}(\mathbf{c})$$

with some penalty function $\mathcal{R} : \mathbb{C}^N \to \mathbb{R}$.

The most common choice for $\mathcal{R}$ when using RFM is the ridge penalty $\mathcal{R}(\mathbf{c}) = \lambda_m \|\mathbf{c}\|_{\ell^2}^2$ which leads to the random feature ridge regression (RFRR) problem [15, 26, 33, 40, 42]. In [42], it was shown that for a function f in an RKHS, using $N = \mathcal{O}(m^{\frac{1}{2}} \log m)$ number of random features is sufficient to achieve a test error of $\mathcal{O}(m^{-\frac{1}{2}})$ when training the output layer of the RFRR problem. In general, to achieve a risk bound that scales like $\mathcal{O}(N^{-1} + m^{-\frac{1}{2}})$ using the RFRR problem over the RKHS, properties of the spectrum of the kernel operator must be known [1, 15]. In practice, the technical assumptions on the spectrum and its decay may be hard to verify for a given problem or dataset.

When the ridge parameter $\lambda_m \to 0^+$, the RFRR becomes the min-norm interpolation problem (also referred to as ridge-less regression) and has received recent attention in several different settings [3–5, 22, 27, 33, 34, 44]. A detailed analysis of both the kernel ridge regression and the RFRR problems in terms of the dimensional parameters d, m, and N is provided in [12, 33], where one conclusion indicates that the min-norm interpolators are the optimal approximations among all kernel methods

when $N \gg m$, i.e. in the overparameterized setting. However, noise, outliers, and model misfits often necessitate the use of a penalty in the regression problem.

Since the risk bounds scale is like N^{-1}, in order to have a small error, one must choose a large number of features. In addition, it has been observed that the global minimizer of the risk using the RFRR problem as a function of the ratio $\frac{N}{m}$ is achieved for values $\frac{N}{m} \gg 1$ [33, 34], that is, the lower risk solutions occur in the very overparameterized limit. While the risk analysis indicates that large N is needed, the complexity of evaluating an RFM in the overpameterized setting scales linearly with the number of features N, but large N can limit the usefulness for scientific problems or many query applications. Thus, an alternative approach is to use sparsity-promoting penalties (or algorithms) to obtain sparse or low-complexity models in the overparameterized setting.

In [21], an ℓ^1 basis pursuit denoising problem was used to train RFM from limited (and noisy) measurements. Similar to the RFRR problem, it was shown that when the sparsity level $s = N$ and the number of measurements $m = \mathcal{O}(N \log(N))$, the risk is bounded by $\mathcal{O}(N^{-1} + m^{-\frac{1}{2}})$ [12]. When the "true" values of the final weight layer $\mathbf{c}$ are compressible, then s can be much smaller than N to achieve similar bounds [12, 21]. In [49], a related algorithm based on the LASSO problem was used to iteratively add a sparse number of random features to the trained RFM. While the ℓ^1 based approaches lead to good generalization bounds and other theoretical properties, their optimization often comes at a higher computational cost than the ridge regression problem. In [47], an iterative pruning approach (related to an ℓ^0 penalized problem) is shown to perform well on several synthetic and real data examples, in particular, for high-dimensional approximation problems with the inherent low-order structure. They connect the sparsity-promoting methods for RFM to the pruning approaches for reducing the model complexity of overparameterized neural networks. Pruning algorithms focus on obtaining small subnetworks with similar accuracy to the full neural network [18, 51] and are based on the lotto ticket hypothesis, where the existence of smaller subnetworks with similar (or better) risk is conjectured [18]. More precisely, the lottery ticket hypothesis said that any randomly initialized, dense neural network (i.e., a random feature network) contains a subnetwork that can match the test accuracy of the original network after training for at most the same number of training iterations. On the other hand, the desired sparsity obtained from the hard thresholding step in our algorithm is essentially a pruning step in order to obtain a subset of features that could lead to better generalization results with a smaller RFM. In this way, we think of thresholding methods as one possible approach to solving the lottery ticket hypothesis.

In this work, we propose the HARFE method, which is a hard ridge-based thresholding algorithm to iteratively obtain a small sub-network for a RFM using a sparsity prior. In addition, we use the sparse random features introduced [21], which restrict each column of the weight matrix $\mathbf{W}$ to have a fixed number of non-zero terms (often only a few non-zero components). This is beneficial when the number of interacting or active variables is low but the ambient dimension is high, see [20, 36, 37]. In particular, for sparse additive modeling and decompositions, the sparse random features represent possible interactions between input variables. Additive models for kernel regression and sparse additive models include the multiple kernel learning algorithms

[2, 19, 48], shrunk additive least squares approximation (SALSA) [25], sparse shrunk additive models (SSAM) [29], component selection and smoothing operator (COSSO) [28], additive model with kernel regularization (KAM) [13], and other related works [37, 45].

1.1 Contributions

Our main algorithmic and modeling contributions are as follows:

- We propose the hard-ridge random feature expansion method (HARFE),[1] which is used to train sparse additive random feature models. This method is related to the lotto ticket hypothesis since the desired sparsity obtained from the hard thresholding step in our algorithm is essentially a pruning step in order to obtain a subset of features that could lead to better generalization results with a smaller RFM, see also [18, 51].
- The sparse ridge regression (SRR) problem and some associated (scalable) algorithms have been studied in [6, 23, 30, 32, 46]. The HARFE algorithm is one way to solve the SRR problem and has guarantees on the convergence when the matrix has standard compressive sensing structures. Specifically, the random feature matrix has a coherence bound [21] and the restricted isometry property [12], thus convergence is given by the results in [9, 10, 16, 31]. We also obtained a generalization bound for our approach based on the proofs from [12, 21].
- Tests on synthetic and real-world datasets validate the proposed method, including the inclusion of sparse random features (i.e. for sparse additive random feature modeling). The method performs comparably or better than other related algorithms as shown in the experiments. In addition, the sparsity priors help to obtain important variable dependencies from the data.

2 Problem statement and algorithm

Throughout the paper, we use bold letters for column vectors (e.g. $\mathbf{x}$) and bold capital letters for matrices (e.g. $\mathbf{A}$). We say that a vector $\mathbf{z}$ is s-sparse if it has at most s nonzero entries. Let $[N]$ denote the set of all positive integers less than or equal to N. We denote the ℓ^p norm of a vector $\mathbf{z}$ by $||\mathbf{z}||_p$. Next, we recall a useful definition.

Definition 2.1 (*Order-q additive functions* [21]) Fix $d, q, K \in \mathbb{N}$ with $1 \leq q \leq d$. A function $f : \mathbb{R}^d \to \mathbb{C}$ is called an order-q additive function with at most K terms if there exist K complex-valued functions $g_1, \ldots, g_K : \mathbb{R}^q \to \mathbb{C}$ such that

$$f(\mathbf{x}) = \frac{1}{K} \sum_{j=1}^{K} g_j(\mathbf{x}|_{\mathcal{S}_j}), \tag{1}$$

where for each $j \in [K]$, $\mathcal{S}_j \subseteq [d]$, $\mathcal{S}_j$ has q distinct indices, and $\mathcal{S}_j \neq \mathcal{S}_{j'}$ for $j \neq j'$. Here $\mathbf{x}|_{\mathcal{S}_j}$ denotes the restriction of $\mathbf{x} \in \mathbb{R}^d$ onto $\mathcal{S}_j$.

[1] The code is available at https://github.com/esha-saha/HARFE.

Note that the class of additive functions in Definition 2.1 is related to sparse additive modeling and multiple kernel learning [2, 13, 19, 20, 25, 28, 29, 36, 37, 48].

We are interested in approximating an unknown high dimensional function $f : \mathbb{R}^d \to \mathbb{C}, d \gg 1$, from a set of m samples $\{(\mathbf{x}_k, y_k)\}_{k=1}^m$ where the inputs $\mathbf{x}_k$ are drawn from an (unknown) probability measure $\mu(\mathbf{x})$ and the output data is (likely) corrupted by noise:

$$y_k = f(\mathbf{x}_k) + e_k, \quad \text{for } k \in [m]. \tag{2}$$

We assume that the noise e_k is a Gaussian random variable or bounded by some constant E, that is, $|e_k| \leq E \ \forall k \in [m]$. In addition, we assume that the target function f is an order-q additive function with $q \ll d$, $K \ll \binom{d}{q}$, and that we do not have prior knowledge on the terms g_j. Using the random feature method [21, 38], we approximate the target function f by:

$$f(\mathbf{x}) \approx f^{\#}(\mathbf{x}) = \mathbf{c}^T \phi(\mathbf{W}^T \mathbf{x}) = \sum_{j=1}^{N} c_j \phi(\langle \mathbf{x}, \boldsymbol{\omega}_j \rangle),$$

where $\mathbf{W} = [\boldsymbol{\omega}_{k,j}] \in \mathbb{R}^{d \times N}$ is a random weight matrix, $\boldsymbol{\omega}_j \in \mathbb{R}^d$ are the column vectors of the matrix $\mathbf{W}$, and $\mathbf{c} \in \mathbb{C}^N$ is the coefficient vector. The random weight matrix $\mathbf{W} \in \mathbb{R}^{d \times N}$ is fixed, while the coefficients $\mathbf{c} \in \mathbb{C}^N$ are trainable. The function $\phi : \mathbb{R} \to \mathbb{R}$ is the nonlinear activation function and can be chosen to be a trigonometric function, the sigmoid function, or the ReLU function. Unless otherwise stated, we use the sine activation function, i.e. $\phi(\cdot) = \sin(\cdot)$. This model is a two-layer neural network with the weights in the hidden layer being randomized but not trainable and thus the training problem relies on learning the coefficient vector $\mathbf{c}$. Theoretically, the random feature method has been shown to be comparable with shallow networks in terms of theoretical risk bounds [38–40, 42] where the population risk is defined as

$$R(f^{\#}) := ||f - f^{\#}||^2_{L^2(d\mu)} = \int_{\mathbb{R}^d} |f(\mathbf{x}) - f^{\#}(\mathbf{x})|^2 d\mu(\mathbf{x}) \tag{3}$$

which is also the L^2 squared error between the true function and its approximation. Suppose the entries of the random weight matrix $\mathbf{W}$ are i.i.d. random variables generated by the (one-dimensional) probability function $\rho(\boldsymbol{\omega}), \boldsymbol{\omega}_{k,j} \sim \rho(\boldsymbol{\omega})$. Let $\mathbf{A} \in \mathbb{C}^{m \times N}$ be the random feature matrix whose entries are defined as $a_{k,j} = \phi(\langle \mathbf{x}_k, \boldsymbol{\omega}_j \rangle)$. Then the general regression problem is to solve the following optimization problem:

$$\min_{\mathbf{c} \in \mathbb{C}^N} ||\mathbf{A}\mathbf{c} - \mathbf{y}||_2^2, \tag{4}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T$.

For sparse additive modeling, since f is an order-q function, each entry of the random feature matrix $\mathbf{A}$ should only depend on q entries of the input data $\mathbf{x}_k$.

Therefore, we can instead generate a sparse random matrix $\mathbf{W}$ where each column of $\mathbf{W}$ has at most q nonzero entries following the probability function $\rho(\boldsymbol{\omega})$ [21]. One such way to generate $\mathbf{W}$ is to first generate N random vectors $\mathbf{v}^{(j)} = (v_1^{(j)}, \ldots, v_q^{(j)})^T$ in $\mathbb{R}^q$ and use a random embedding that assigns $\mathbf{v}^{(j)}$ to $\boldsymbol{\omega}_j$, where $\boldsymbol{\omega}_j = (0, 0, \ldots, 0, v_1^{(j)}, 0, \ldots, v_2^{(j)}, 0, \ldots, v_q^{(j)}, 0, \ldots, 0)^T$. In particular, for each j, we select a subset of q indices from $[d]$ uniformly at random and then sample each nonzero entry using $\rho(\boldsymbol{\omega})$. The sparse random matrix $\mathbf{W}$ can also be obtained as $\mathbf{W} = \widetilde{\mathbf{W}} \odot \mathbf{M}$, where $\widetilde{\mathbf{W}} \in \mathbb{R}^{d\times N}$ is a dense matrix whose entries are sampled from $\rho(\boldsymbol{\omega})$, the mask $\mathbf{M} \in \mathbb{R}^{d\times N}$ is a sparse matrix whose non-zero entries are one and each column of $\mathbf{M}$ has q non-zero entries, and $\odot$ denotes the element-wise multiplication. Using this formulation, the general sparse regression problem becomes

$$\text{find } \mathbf{c} \in \mathbb{C}^N \text{ such that } ||\mathbf{A}\mathbf{c} - \mathbf{y}||_2 \leq \epsilon\sqrt{m} \quad \text{and} \quad \mathbf{c} \quad \text{is sparse}, \tag{5}$$

where ϵ is the parameter related to the noise level. The motivation for sparsity in $\mathbf{c}$ is due to the assumption that $K \ll \binom{d}{q}$, thus not all index subsets are needed.

In order to solve the sparse random feature regression problem, we propose a new greedy algorithm named hard-ridge random feature expansion (HARFE), which uses a hard thresholding pursuit (HTP) like algorithm to solve the random feature ridge regression problem. Specifically, we learn $\mathbf{c}$ from the following minimization problem:

$$\min_{\mathbf{c}\in\mathbb{C}^N} \|\mathbf{A}\mathbf{c} - \mathbf{y}\|_2^2 + m\lambda||\mathbf{c}||_2^2 \quad \text{such that} \quad \mathbf{c} \text{ is } s \text{ -sparse}, \tag{6}$$

where $\lambda > 0$ is the regularization parameter. Equation (6) can be rewritten as

$$\min_{\mathbf{c}\in\mathbb{C}^N} \|\mathbf{B}\mathbf{c} - \tilde{\mathbf{y}}||_2^2 \quad \text{such that } \mathbf{c} \text{ is } s\text{-sparse}, \tag{7}$$

where $\mathbf{B} = \begin{bmatrix} \mathbf{A} \\ \sqrt{m\lambda}\mathbf{I}_N \end{bmatrix} \in \mathbb{C}^{(m+N)\times N}$ and $\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{m+N}$. To solve (7), we first start with an s-sparse vector $\mathbf{c}^0 \in \mathbb{C}^N$ (typically taken as $\mathbf{c}^0 = 0$) and update based on the HTP approach

$$\begin{aligned} S^{n+1} &= \{\text{indices of } s \text{ largest (in magnitude) entries of } \mathbf{c}^n + \mu\mathbf{B}^*(\mathbf{y} - \mathbf{B}\mathbf{c}^n)\} \\ \mathbf{c}^{n+1} &= \text{argmin}\{\|\tilde{\mathbf{y}} - \mathbf{B}\mathbf{c}\|_2^2, \quad \text{supp}(\mathbf{c}) \subseteq S^{n+1}\} \end{aligned} \tag{8}$$

where $\mu > 0$ is the step-size and s is a user defined parameter. The idea is to solve for the coefficients using a much smaller number of model terms. The subset S given by the indices of the s largest entries of one gradient descent step applied on the vector $\mathbf{c}$ is a good candidate for the support set of $\mathbf{c}$. The HTP algorithm iterates between these two steps and leads to a stable and robust reconstruction of sparse vectors depending on the RIP constant. The Gram matrix $\mathbf{B}^*\mathbf{B}$ is computed directly based on the ridge problem

$$\mathbf{c}^n + \mu\mathbf{B}^*(\tilde{\mathbf{y}} - \mathbf{B}\mathbf{c}^n) = (1 - m\mu\lambda)\mathbf{c}^n + \mu\mathbf{A}^*(\mathbf{y} - \mathbf{A}\mathbf{c}^n).$$

Algorithm 1 Hard-Ridge Random Feature Expansion (HARFE)

Require: Samples $\{\mathbf{x}_k, y_k\}_{k=1}^m$, non-linear function ϕ, number of features N, sparsity level s, random weight sparsity q, step size μ, regularization parameter λ, convergence threshold ϵ and total number of iterations `tot_iter`.

Draw (q-sparse) N random weights $\boldsymbol{\omega}_j$ whose non-zero entries are sampled from $\rho(\boldsymbol{\omega})$.

Construct the random feature matrix $\mathbf{A} = [\phi(\langle \mathbf{x}_k; \boldsymbol{\omega}_j \rangle)] \in \mathbb{C}^{m \times N}$.

Ensure:

Initialization: Start with s-sparse $\mathbf{c}^0 \in \mathbb{C}^N$ ($\mathbf{c}^0 = 0$), $n = 0$

while (Relative Residual$> \epsilon$) or (`n<tot_iter`) **do**

$\tilde{\mathbf{c}}^{n+1} \leftarrow (1 - m\mu\lambda)\mathbf{c}^n + \mu\mathbf{A}^*(\mathbf{y} - \mathbf{A}\mathbf{c}^n)$

idx $\leftarrow$ indices of s largest entries of $\tilde{\mathbf{c}}^{n+1}$ ▷ Choose the subset of features S^{n+1}

$\bar{\mathbf{A}} \leftarrow \mathbf{A}[:, \text{idx}]$

$\mathbf{c}^{n+1}[[N] \setminus \text{idx}] = 0$

$\mathbf{c}^{n+1}[\text{idx}] \leftarrow \text{argmin}\{||\mathbf{y} - \bar{\mathbf{A}}\mathbf{c}||_2^2 + m\lambda||\mathbf{c}||_2^2\} = (\bar{\mathbf{A}}^*\bar{\mathbf{A}} + m\lambda\mathbf{I}_s)^{-1}\bar{\mathbf{A}}^*\mathbf{y}$

$n = n + 1$

end while

return Sparse vector $\mathbf{c} = [c_1, c_2, \ldots, c_N]$ such that: $f^\sharp(\mathbf{x}) = \sum_{j=1}^{N} c_j \phi(\mathbf{x}; \boldsymbol{\omega}_j)$.

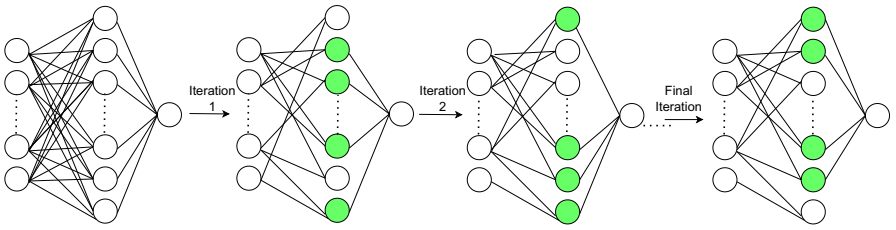

Fig. 1 Schematic representation of HARFE. The active nodes at each iteration is given in green

The approach is summarized in Algorithm 1. The relative residual at the iterate $\mathbf{c}^n$ is defined as

$$\text{Relative Residual} = \frac{||\mathbf{A}\mathbf{c}^n - \mathbf{y}||_2}{||\mathbf{y}||_2}.$$

One of the motivations for including the ridge penalty is that for random feature regression, the sparsity level s can be large and thus the least squares step in (8) may be ill-conditioned. Numerically, we observed that even a small non-zero value of λ can be beneficial for ensuring convergence and good generalization. In Fig. 1, we provide a schematic representation of the sequence generated by the algorithm, namely, over each step a sparse subset of nodes is obtained until a final configuration is achieved.

3 Theoretical discussion

The error produced by the HARFE algorithm can be established by extending the results on the HTP algorithm to include the ridge regression term and by leveraging bounds on the restricted isometry constant for this type of random feature matrix. Recall that given an integer $s \in [N]$, the s-th restricted isometry constant of a matrix

$\mathbf{A} \in \mathbb{C}^{m\times N}$, denoted by $\delta_s(\mathbf{A})$, is the smallest non-negative δ such that

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2$$

holds for all s-sparse $\mathbf{x} \in \mathbb{C}^N$ [17]. Let $\kappa_{1,s}(\mathbf{x})$ denote the ℓ^1 distance to the best s-term approximation of $\mathbf{x}$ defined by

$$\kappa_{1,s}(\mathbf{x}) = \inf\left\{\|\mathbf{z}-\mathbf{x}\|_1 : \mathbf{z} \in \mathbb{C}^n,\ \mathbf{z} \text{ is } s-\text{sparse}\right\}.$$

The value $\kappa_{1,s}(\mathbf{x})$ provides a measure for the compressibility of the vector $\mathbf{x}$ with respect to the ℓ^1 norm and is obtained by setting all but the s-largest in magnitude entries to zero.

The following result is restricted to the case when $q = d$ and $\mu = 1$. The $q \leq d$ case follows from similar arguments [21]. From [17], the theorem below can be extended trivially for any step size μ by scaling the matrix $\mathbf{A}$ and the vector $\mathbf{y}$ with the ratio $\sqrt{\mu}$.

Theorem 3.1 (Convergence of the iterates of HARFE) *Let the data $\{\mathbf{x}_k\}_{k\in[m]}$ be drawn from $\mathcal{N}(\mathbf{0}, \gamma^2\mathbf{I}_d)$, the weights $\{\boldsymbol{\omega}_j\}_{j\in[N]}$ be drawn from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$, and the random feature matrix $\mathbf{A} \in \mathbb{C}^{m\times N}$ be defined component-wise by $a_{k,j} = \exp(i\langle\mathbf{x}_k, \boldsymbol{\omega}_j\rangle)$. Denote the ℓ_2-regularization parameter by λ and the sparsity level by s. If*

$$\begin{aligned} m &\geq C_1\,(1+\lambda)^{-2}\,s\,\log(\delta^{-1}),\\ \frac{m}{\log(3m)} &\geq C_2\,(1+\lambda)^{-1}\,s\,\log^2(6s)\log\left(\frac{N}{9\log(2m)}+3\right),\\ \frac{\sqrt{\delta}}{6\sqrt{3}}\,(1+\lambda)\,(4\gamma^2\sigma^2+1)^{\frac{d}{4}} &\geq N, \end{aligned}$$

where C_1 and C_2 are universal positive constants, then with probability at least $1-2\delta$, for all $\mathbf{c} \in \mathbb{C}^N$ and $\mathbf{e} \in \mathbb{C}^m$ with $\mathbf{y} = \mathbf{A}\mathbf{c}+\mathbf{e}$, the sequence $\mathbf{c}^n$ defined by the Algorithm 1 with $\mathbf{c}^0 = \mathbf{0}$, using $2s$ instead of s in the algorithm, satisfies

$$\|\mathbf{c}^n-\mathbf{c}\|_2 \leq 2\beta^n\|\mathbf{c}\|_2 + \frac{D_1}{\sqrt{s}}\kappa_{1,s}(\mathbf{c}) + D_2\sqrt{\frac{m^{-1}\|\mathbf{y}-\mathbf{A}\mathbf{c}\|_2^2+\lambda\|\mathbf{c}\|_2^2}{1+\lambda}}$$

for all $n \geq 0$ where the constants $\beta \in (0,1)$, $D_1, D_2 > 0$ depend only on $\delta_{6s}(\mathbf{B})$. The matrix $\mathbf{B}$ is given by $\mathbf{B} = (m+m\lambda)^{-\frac{1}{2}}\begin{bmatrix}\mathbf{A}\\ \sqrt{m\lambda}I_N\end{bmatrix} \in \mathbb{C}^{(m+N)\times N}$.

Proof The ridge regression problem:

$$\min_{\mathbf{c}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{c}-\mathbf{y}\|_2^2 + m\lambda\|\mathbf{c}\|_2^2 \tag{9}$$

can be written in the form:

$$\min_{\mathbf{z}' \in \mathbb{R}^N} \|\mathbf{B}\mathbf{z}' - \tilde{\mathbf{y}}\|_2^2 \tag{10}$$

where $\mathbf{B} = (m + m\lambda)^{-\frac{1}{2}} \begin{bmatrix} \mathbf{A} \\ \sqrt{m\lambda} I_N \end{bmatrix} \in \mathbb{C}^{(m+N)\times N}$ and $\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{m+N}$. For Eq. (10), the error is defined as $\tilde{\mathbf{e}} = \tilde{\mathbf{y}} - \mathbf{B}\mathbf{z}'$ which is equivalent to

$$\tilde{\mathbf{e}} = \begin{bmatrix} \mathbf{e} \\ -\sqrt{m\lambda}\,\mathbf{c} \end{bmatrix} \in \mathbb{C}^{m+N}.$$

If $\mathbf{c}$ is the minimizer of (9) and $\mathbf{z}'$ is the minimizer of (10), then $\mathbf{z}' = (m + m\lambda)^{\frac{1}{2}}\,\mathbf{c}$. The scaling is needed for the matrix to satisfy a restricted isometry property.

To estimate the restricted isometry constant of $\mathbf{B}$, we first bound the restricted isometry constant of $m^{-\frac{1}{2}}\mathbf{A}$. By Theorem A.1 and the assumptions, if

$$\begin{aligned} m &\geq 6C_1\eta_1^{-2}\, s\, \log(\delta^{-1}) \\ \frac{m}{\log(3m)} &\geq 6C_2\eta_2^{-2}\, s\, \log^2(6s) \log\left(\frac{N}{9\log(2m)} + 3\right) \\ \sqrt{\delta}\,\eta_1\,(4\gamma^2\sigma^2 + 1)^{\frac{d}{4}} &\geq N, \end{aligned}$$

where C_1 and C_2 are universal positive constants, then with probability at least $1 - 2\delta$, the $6s$-restricted isometry constant is bounded by

$$\delta_{6s}(\mathbf{A}) < 3\eta_1 + \eta_2^2 + \sqrt{2}\eta_2,$$

or equivalently

$$\left\| m^{-1}\mathbf{A}_S^*\mathbf{A}_S - \mathbf{I}_S \right\|_{2\to 2} < 3\eta_1 + \eta_2^2 + \sqrt{2}\eta_2,$$

for all $S \subset [N]$ with $|S| = 6s$. Therefore, the $6s$-restricted isometry constant of B satisfies

$$\begin{aligned} \left\| \mathbf{B}_S^*\mathbf{B}_S - \mathbf{I}_S \right\|_{2\to 2} &= \left\| (m + m\lambda)^{-1} \left(\mathbf{A}_S^*\mathbf{A}_S + m\lambda\mathbf{I}_S\right) - \mathbf{I}_S \right\|_{2\to 2} \\ &= \frac{1}{1+\lambda} \left\| m^{-1}\mathbf{A}_S^*\mathbf{A}_S - \mathbf{I}_S \right\|_{2\to 2} \\ &< \frac{3\eta_1 + \eta_2^2 + \sqrt{2}\eta_2}{1+\lambda}. \end{aligned}$$

Setting the parameters to $\eta_1 = \frac{1+\lambda}{6\sqrt{3}}$ and $\eta_2 = \frac{\sqrt{1+\lambda}}{4\sqrt{3}}$, then $\left\| \mathbf{B}_S^*\mathbf{B}_S - \mathbf{I}_S \right\|_{2\to 2} < \frac{1}{\sqrt{3}}$ if

$$m \geq 648C_1\,(1+\lambda)^{-2}\, s\, \log(\delta^{-1})$$

$$\frac{m}{\log(3m)} \geq 288C_2\,(1+\lambda)^{-1}\,s\,\log^2(6s)\log\left(\frac{N}{9\log(2m)}+3\right)$$
$$\frac{\sqrt{\delta}}{6\sqrt{3}}\,(1+\lambda)\,(4\gamma^2\sigma^2+1)^{\frac{d}{4}} \geq N.$$

By Eq. (23) of Theorem A.2, the sequence generated by the hard thresholding pursuit algorithm produces a solution with the following bound

$$\|\mathbf{z}'^n - \mathbf{z}'\|_2 \leq 2\beta^n\|\mathbf{z}'\|_2 + \frac{C}{\sqrt{s}}\kappa_{1,s}(\mathbf{z}') + D\|\tilde{\mathbf{e}}\|_2, \tag{11}$$

for all $n \geq 0$ where the constants $\beta \in (0,1), C, D > 0$ depend only on $\delta_{6s}(\mathbf{B})$. Transforming the variables to the original variables yields the following bound

$$\|\mathbf{c}^n - \mathbf{c}\|_2 \leq 2\beta^n\|\mathbf{c}\|_2 + \frac{C}{\sqrt{s}}\kappa_{1,s}(\mathbf{c}) + D\,(1+\lambda)^{-\frac{1}{2}}\,m^{-\frac{1}{2}}\,\|\tilde{\mathbf{e}}\|_2 \tag{12}$$
$$\leq 2\beta^n\|\mathbf{c}\|_2 + \frac{C}{\sqrt{s}}\kappa_{1,s}(\mathbf{c}) + D\,(1+\lambda)^{-\frac{1}{2}}\,m^{-\frac{1}{2}}\sqrt{\|\mathbf{e}\|_2^2 + m\lambda\|\mathbf{c}\|_2^2} \tag{13}$$
$$\leq 2\beta^n\|\mathbf{c}\|_2 + \frac{C}{\sqrt{s}}\kappa_{1,s}(\mathbf{c}) + D\sqrt{\frac{m^{-1}\|\mathbf{e}\|_2^2 + \lambda\|\mathbf{c}\|_2^2}{1+\lambda}}, \tag{14}$$

which concludes the proof. □

It is worth noting that, in practice, $\lambda > 0$ is small, i.e. we would like $m\lambda = \mathcal{O}(1)$. Thus the third term in the iterative bound is smaller than the second term and does not contribute significantly to the overall error.

Theorem 3.2 (Risk bound for HARFE) *Let the data $\{\mathbf{x}_k\}_{k\in[m]}$ be drawn from $\mathcal{N}(\mathbf{0}, \gamma^2\mathbf{I}_d)$, the weights $\{\boldsymbol{\omega}_j\}_{j\in[N]}$ be drawn from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$, and the random feature matrix $\mathbf{A} \in \mathbb{C}^{m\times N}$ be defined component-wise by $a_{k,j} = \exp(i\langle\mathbf{x}_k, \boldsymbol{\omega}_j\rangle)$. Denote the ℓ_2 regularization parameter by λ, accuracy parameter by $\epsilon > 0$, and the sparsity level by s. If the assumptions of Theorem 3.1 are satisfied, then with probability at least $1-2\delta$, the following risk bounds hold:*

$$R(f^{\#}) \leq \|f\|_\rho\left(\epsilon + D\sqrt{\frac{\lambda}{N}}\left(3^{-\frac{1}{4}} + \left(2m\log\left(\frac{1}{\delta}\right)\right)^{\frac{1}{4}}\right)\right)$$
$$+ C\sqrt{\frac{1+\lambda}{s}}\left(3^{-\frac{1}{4}} + N^{\frac{1}{2}}\left(2m\log\left(\frac{1}{\delta}\right)\right)^{\frac{1}{4}}\right)\kappa_{s,1}(\|\mathbf{c}^\star\|)$$
$$+ DE\left(3^{-\frac{1}{4}}m^{-\frac{1}{2}} + N^{\frac{1}{2}}\left(\frac{2}{m}\log\left(\frac{1}{\delta}\right)\right)^{\frac{1}{4}}\right).$$

Proof We use the L^2 norm, which can be decomposed into two parts using the triangle inequality:

$$||f - f^\sharp||_{L^2(d\mu)} \leq ||f - f^\star||_{L^2(d\mu)} + ||f^\star - f^\sharp||_{L^2(d\mu)}.$$

The approximation $f^\sharp$ is defined in Eq. (18) and the best ϕ-based approximation $f^\star$ is given in Eq. (19).

Following the proof in Section 6 of [12], if $N \geq \frac{1}{\epsilon^2}(1 + \sqrt{2\log(\frac{1}{\delta})})^2$, then with probability at least $1 - \delta$, we have

$$||f - f^\star||_{L^2(d\mu)} \leq \epsilon ||f||_\rho. \tag{15}$$

We use McDiarmid's Inequality to bound $||f^\star - f^\sharp||_{L^2(d\mu)}$, arguing similarly as in Lemma 2 from [21] or Section 6 of [12]. Let $\{\mathbf{z}_j\}_{j\in[m]}$ be i.i.d. random variables sampled from the distribution μ which are independent from the sampled points $\{\mathbf{x}_j\}_{j\in[m]}$ and the frequencies $\{\boldsymbol{\omega}\}_{k\in[N]}$. This independence assumption makes $\{z_j\}_{j\in[m]}$ also independent of coefficients $\mathbf{c}^\sharp$ and $\mathbf{c}^\star$ and thus allows for the following argument. Let v be a random variable defined by

$$v(\mathbf{z_1}, \ldots, \mathbf{z_m}) = \|f^\star - f^\sharp\|^2_{L^2(d\mu)} - \frac{1}{m}\sum_{j=1}^{m} |f^\star(\mathbf{z}_j) - f^\sharp(\mathbf{z}_j)|^2.$$

Then $\mathbb{E}_{\mathbf{z}}[v] = 0$ as

$$\mathbb{E}_{\mathbf{z}}[|f^\star(\mathbf{z}_j) - f^\sharp(\mathbf{z}_j)|^2] = \mathbb{E}_{\mathbf{z_1},\ldots,\mathbf{z_m}}[|f^\star(\mathbf{z}_j) - f^\sharp(\mathbf{z}_j)|^2] = \|f^\star - f^\sharp\|^2_{L^2(d\mu)}.$$

We perturb the $k-$th component of v to get,

$$\begin{aligned} &|v(\mathbf{z_1}, \ldots, \mathbf{z_k}, \ldots, \mathbf{z_m}) - v(\mathbf{z_1}, \ldots, \tilde{\mathbf{z}}_\mathbf{k}, \ldots, \mathbf{z_m})| \\ &\leq \frac{1}{m}\left||f^\star(\mathbf{z}_k) - f^\sharp(\mathbf{z}_k)|^2 - |f^\star(\tilde{\mathbf{z}}_k) - f^\sharp(\tilde{\mathbf{z}}_k)|^2\right|. \end{aligned}$$

Using Cauchy–Schwarz inequality, for any $\mathbf{z}$, we have,

$$|f^\star(\tilde{\mathbf{z}}_k) - f^\sharp(\tilde{\mathbf{z}}_k)|^2 \leq N\|\tilde{\mathbf{c}}^\star - \tilde{\mathbf{c}}^\sharp\|_2^2,$$

which holds since $|\phi(\mathbf{z}, \boldsymbol{\omega})| = 1$. Hence,

$$|v(\mathbf{z_1}, \ldots, \mathbf{z_k}, \ldots, \mathbf{z_m}) - v(\mathbf{z_1}, \ldots, \tilde{\mathbf{z}}_\mathbf{k}, \ldots, \mathbf{z_m})| \leq \frac{2N}{m}\|\tilde{\mathbf{c}}^\star - \tilde{\mathbf{c}}^\sharp\|_2^2 := \Delta.$$

Therefore, we can apply McDiarmid's inequality to the random variable v, i.e. $P_{\mathbf{z}}(v - \mathbb{E}_{\mathbf{z}}[v] \geq t) \leq \exp(-\frac{2t^2}{m\Delta^2})$ where $t := \Delta\sqrt{\frac{m}{2}\log\left(\frac{1}{\delta}\right)}$. Following the results

from Theorem 3.1, we have that $\delta_{6s}(\mathbf{B}) < \frac{1}{\sqrt{3}}$ (the matrix $\mathbf{B}$ is as obtained in Eq. (10)), then with s replaced by $2s$, with probability at least $1-3\delta$ (2δ for the coherence bound and δ from the 25), we have:

$$\begin{aligned}\|f^\star - f^\sharp\|^2_{L^2(d\mu)} &\leq \frac{1}{m}\sum_{j=1}^m |f^\star(\mathbf{z}_j) - f^\sharp(\mathbf{z}_j)|^2 + N\left(\sqrt{\frac{2}{m}\log\left(\frac{1}{\delta}\right)}\right)\|\tilde{\mathbf{c}}^\star - \tilde{\mathbf{c}}^\sharp\|_2^2 \\ &= \frac{1}{m}\|\tilde{\mathbf{B}}(\tilde{\mathbf{c}}^\star - \tilde{\mathbf{c}}^\sharp)\|_2^2 + N\left(\sqrt{\frac{2}{m}\log\left(\frac{1}{\delta}\right)}\right)\|\tilde{\mathbf{c}}^\star - \tilde{\mathbf{c}}^\sharp\|_2^2 \\ &\leq \frac{1}{\sqrt{3}m}\|(\tilde{\mathbf{c}}^\star - \tilde{\mathbf{c}}^\sharp)\|_2^2 + N\left(\sqrt{\frac{2}{m}\log\left(\frac{1}{\delta}\right)}\right)\|\tilde{\mathbf{c}}^\star - \tilde{\mathbf{c}}^\sharp\|_2^2.\end{aligned}$$

From Eq. (24) of Theorem A.2, $\|(\tilde{\mathbf{c}}^\star - \tilde{\mathbf{c}}^\sharp)\|_2 \leq \frac{C}{\sqrt{s}}\kappa_{s,1}(\tilde{\mathbf{c}}^\star) + D\|\tilde{\mathbf{e}}\|_2$, where $C, D > 0$ depend only on $\delta_{6s}(\mathbf{B})$. Since $\tilde{\mathbf{c}} = \sqrt{m+m\lambda}\mathbf{c}$, transforming to original variables, we have:

$$\begin{aligned}\|f^\star - f^\sharp\|_{L^2(d\mu)} \leq &\left(\frac{1}{\sqrt{3}m} + N\left(\sqrt{\frac{2}{m}\log\left(\frac{1}{\delta}\right)}\right)\right)^{\frac{1}{2}} \\ &\left(\sqrt{m+m\lambda}\frac{C}{\sqrt{s}}\kappa_{s,1}(\|\mathbf{c}^\star\|) + D\sqrt{\|\mathbf{e}\|^2 + m\lambda\|\mathbf{c}^\star\|_2^2}\right) \qquad (16)\end{aligned}$$

Note $|\mathbf{c}_k^\star| = \frac{1}{N}\left|\frac{\alpha(\boldsymbol{\omega}_k)}{\rho(\omega)_k}\right| \leq \frac{1}{N}\|f\|_\rho$ and $\|\mathbf{e}\|_2 \leq E$. Thus, combining Eqs. (25) and (16) yields

$$\begin{aligned}\|f - f^\sharp\|_{L^2(d\mu)} &\leq \epsilon\|f\|_\rho + \left(\frac{1}{\sqrt{3}m} + N\left(\sqrt{\frac{2}{m}\log\left(\frac{1}{\delta}\right)}\right)\right)^{\frac{1}{2}} \\ &\quad\left(\sqrt{m+m\lambda}\frac{C}{\sqrt{s}}\kappa_{s,1}(\|\mathbf{c}^\star\|) + D\sqrt{E^2 + m\lambda N^{-1}\|f\|_\rho^2}\right) \\ &\leq \epsilon\|f\|_\rho + \left(3^{-\frac{1}{4}}m^{-\frac{1}{2}} + N^{\frac{1}{2}}\left(\frac{2}{m}\log\left(\frac{1}{\delta}\right)\right)^{\frac{1}{4}}\right) \\ &\quad\left(\sqrt{m}\sqrt{1+\lambda}\frac{C}{\sqrt{s}}\kappa_{s,1}(\|\mathbf{c}^\star\|) + D(E + \sqrt{m\lambda}N^{-\frac{1}{2}}\|f\|_\rho\right) \\ &= \|f\|_\rho\left(\epsilon + D\sqrt{\frac{\lambda}{N}}\left(3^{-\frac{1}{4}} + \left(2m\log\left(\frac{1}{\delta}\right)\right)^{\frac{1}{4}}\right)\right) \\ &\quad + C\sqrt{\frac{1+\lambda}{s}}\left(3^{-\frac{1}{4}} + N^{\frac{1}{2}}\left(2m\log\left(\frac{1}{\delta}\right)\right)^{\frac{1}{4}}\right)\kappa_{s,1}(\|\mathbf{c}^\star\|) \\ &\quad + DE\left(3^{-\frac{1}{4}}m^{-\frac{1}{2}} + N^{\frac{1}{2}}\left(\frac{2}{m}\log\left(\frac{1}{\delta}\right)\right)^{\frac{1}{4}}\right).\end{aligned}$$

□

Theorem 3.1 highlights a theoretical purpose of $\lambda > 0$ in terms of convergence. The constants $D_1, D_2 > 0$ in Theorem 3.1 depend on δ_{6s} (**B**). As λ approaches zero, the value of δ_{6s} (**B**) approaches the larger value of δ_{6s} (A) and thus β increases, which can lead to slower convergence. As λ becomes large, the solution approaches zero and the error bounds in Theorem 3.1 become trivial. In practice, we found that a small non-zero value is useful for convergence and for mitigating the effects of noise and outliers.

Theorem 3.1 is stated for any vector $\mathbf{c}$. We can consider two potential vectors $\mathbf{c}$ depending on the scaling of N and m. First, if N is sufficiently large, then by the results of [12, 21] the matrix $\mathbf{A}$ will be well-conditioned with high probability (for any $\lambda > 0$) and thus there exists a $\mathbf{c}$ such that $\mathbf{e} = \mathbf{0}$. Therefore, for small $\lambda > 0$ the relative error is dominated by the compressibility of $\mathbf{c}$:

$$\frac{\|\mathbf{c}^n - \mathbf{c}\|_2}{\|\mathbf{c}\|_2} \leq 2\beta^n + \frac{C}{\sqrt{s}} \frac{\kappa_{1,s}(\mathbf{c})}{\|\mathbf{c}\|_2} + D\sqrt{\frac{\lambda}{1+\lambda}}. \tag{17}$$

In this setting, the HARFE algorithm can be seen as a pruning approach that generates a subnetwork with s connections that is an approximation to the full N-parameter network, see [18, 51].

Alternatively, we can consider the function approximation results found in [21, 38–40]. Suppose we are given a probability density ρ used to sample the entries of the random weights $\boldsymbol{\omega} \in \mathbb{R}^d$ and a function $\phi : \mathbb{R}^{2d} \to \mathbb{C}$. A function $f \in \mathcal{F}(\phi, \rho)$ if $f : \mathbb{R}^d \to \mathbb{C}$ has finite ρ-norm with respect to ϕ defined by

$$\mathcal{F}(\phi, \rho) = \left\{ f(\mathbf{x}) = \int_{\mathbb{R}^d} \alpha(\boldsymbol{\omega}) \phi(\mathbf{x}; \boldsymbol{\omega})\, d\boldsymbol{\omega} \;\middle|\; \|f\|_\rho := \sup_{\boldsymbol{\omega}} \left| \frac{\alpha(\boldsymbol{\omega})}{\rho(\boldsymbol{\omega})} \right| < \infty \right\},$$

where $\rho(\boldsymbol{\omega}) = \rho(\omega_1) \ldots \rho(\omega_d)$. The random feature approximation of $f \in \mathcal{F}(\phi, \rho)$ is denoted by $f^\sharp$ and defined as

$$f^\sharp(\mathbf{x}) = \sum_{j=1}^{N} c_j^\sharp \, \phi(\mathbf{x}, \boldsymbol{\omega}_j), \tag{18}$$

where the weights $\{\boldsymbol{\omega}_j\}_{j \in [N]}$ are sampled i.i.d. from the density ρ. Following [38–40], the best ϕ-based approximation of $f \in \mathcal{F}(\phi, \rho)$ is given by $f^\star$

$$f^\star(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^{N} \frac{\alpha(\boldsymbol{\omega}_j)}{\rho(\boldsymbol{\omega}_j)} \phi(\mathbf{x}, \boldsymbol{\omega}_j), \tag{19}$$

where the coefficients with respect to the random features are defined as $c_j^\star := \frac{\alpha(\boldsymbol{\omega}_j)}{N\rho(\boldsymbol{\omega}_j)}$ for all $j \in [N]$ and thus $\|c^\star\|_2 \leq N^{-\frac{1}{2}} \|f\|_\rho$. For example, let ρ be the density associated with $\mathcal{N}(0, \sigma^2)$ and assume that the conditions of Theorem 3.1 hold. In this

setting, it was shown in [21] that $\|\mathbf{e}^\star\|_\infty = \|A\mathbf{c}^\star - \mathbf{y}\|_\infty \leq \epsilon \|f\|_\rho$, where

$$\epsilon := \frac{1}{\sqrt{N}} \left(1 + 4\gamma\sigma d \sqrt{1 + \sqrt{\frac{12}{d} \log \frac{m}{\delta}}} + \sqrt{\frac{1}{2} \log \left(\frac{1}{\delta} \right)} \right).$$

Therefore, the bound in Theorem 3.1 becomes

$$\|\mathbf{c}^n - \mathbf{c}^\star\|_2 \leq \left(2\beta^n N^{-\frac{1}{2}} + \frac{C}{\sqrt{s}} \left(1 - sN^{-1} \right) + D \sqrt{\frac{\epsilon^2 + \lambda N^{-1}}{1 + \lambda}} \right) \|f\|_\rho, \quad (20)$$

which scales like $N^{-\frac{1}{2}}$. Equation (20) could be refined, since in multiple places an ℓ^2 bound is replaced by an ℓ^∞ norm (noting that $\|f\|_\rho$ is essentially an infinity-like norm).

For the HARFE results in the following two sections, we observed that the value of μ does not have a significant impact on the generalization. Therefore, we set $\mu = 0.1$ for all experiments, which can be shown to produce a convergent sequence by extending the proofs found in [16]. In addition, we fix the sparsity ratio (s/N) to be 5% or 10%. Since the parameter λ depends on the dataset and the noise level, it should be tuned, for example, using cross-validation. In our experiments, we found that the optimal regularization parameter can range from 10^{-12} to 10^{-1} depending on the input data (since the data is not normalized). Thus, we optimize our results over a set of possible values for λ in order to obtain good generalization.

4 Numerical results on synthetic data

In this section, we test Algorithm 1 for approximating sparse additive functions including the benchmark examples discussed in [7, 8, 21, 35, 37]. The experiments show that the HARFE model outperforms several existing methods in terms of testing errors.

The step-size parameter is set to $\mu = 0.1$ for all experiments. Other hyperparameters will be specified for each experiment. The relative test error is calculated as

$$\text{Rel}(f, f^\sharp) = \sqrt{\frac{\sum_{\mathbf{x} \in X_{\text{test}}} |f(\mathbf{x}) - f^\sharp(\mathbf{x})|^2}{\sum_{\mathbf{x} \in X_{\text{test}}} |f(\mathbf{x})|^2}}, \quad (21)$$

and the mean-squared test error is defined as

$$\text{MSE}(f, f^\sharp) = \frac{1}{|X_{\text{test}}|} \sum_{\mathbf{x} \in X_{\text{test}}} |f(\mathbf{x}) - f^\sharp(\mathbf{x})|^2 \quad (22)$$

where f is the target function and $f^\sharp$ is the trained function.

Table 1 Relative test errors (as percentage) for approximating various nonlinear functions using different q values

d	q	$\frac{1}{\sqrt{1+\|\mathbf{x}\|_2^2}}$	$\sqrt{1+\|\mathbf{x}\|_2^2}$	$\frac{x_1 x_2}{1+x_3^6}$	$\sum_{i=1}^{d}\exp(-\lvert x_i\rvert)$
		5	5	5	100
SRFE	1	3.30	1.29	102.1	**1.20**
HARFE, $\lambda = 0$	1	3.30	1.40	105.6	1.50
HARFE, $\lambda > 0$	1	3.20	1.00	100	**1.10**
SRFE	3	0.80	1.0	8.0	1.80
HARFE, $\lambda = 0$	3	1.00	**0.25**	**3.20**	2.70
HARFE, $\lambda > 0$	3	0.73	**0.18**	**3.40**	2.01
SRFE	5	**0.56**	1.10	9.24	2.04
HARFE, $\lambda = 0$	5	1.80	1.08	11.20	3.00
HARFE, $\lambda > 0$	5	**0.57**	1.00	7.70	2.20

For each function, the two smallest errors are highlighted (in bold). The ridge paramater λ using in the HARFE approach is set to 10^{-4}, 10^{-10}, 10^{-10}, and 10^{-1} (going left to right). In all experiments, $m_{train} = m_{test} = 500$, $N = 10^4$, $\mathbf{x} \sim \mathcal{U}[-1, 1]^d$, the nonzero entries of $\boldsymbol{\omega}$ are drawn from $\mathcal{N}(0, 1)$ and bias is drawn from $\mathcal{U}[0, 2\pi]$

4.1 Low-order function approximation

In the first example, we show the advantage of using a greedy approach over an ℓ^1 optimization problem and the benefit of the additional ridge term. The input data is sampled from a uniform distribution $\mathcal{U}[-1, 1]^d$ and the activation function is set to $\phi(\cdot) = \sin(\cdot)$. The nonzero entries of the random weights $\boldsymbol{\omega}_j$ are sampled from $\mathcal{N}(0, 1)$. We introduce a set of random bias terms $b_j \in \mathbb{R}$ for $j \in [N]$ so that the random feature matrix is now defined as $a_{k,j} = \sin(\langle \mathbf{x}_k, \boldsymbol{\omega}_j \rangle + b_j)$. The bias is sampled from $\mathcal{U}[0, 2\pi]$ to cover all phase angles. The number of random weights is $N = 10^4$, the sparsity level is $s = 500$, and the number of training and testing data are $m_{\text{train}} = 500$ and $m_{\text{test}} = 500$, respectively. In all experiments in this section, we set the maximal number of iterations for our method to 50.

Table 1shows the median relative error (as a percentage) over 10 randomly generated test sets. In the experiments, we compared the results using $q \in \{1, 3, 5\}$. In each case, the HARFE approach with $\lambda > 0$ is more accurate than HARFE with $\lambda = 0$ and the SRFE [21]. We observe that when the exact order q is known, the error is lower (see Column 5 of Table 1). Although not included in the table, it is worth mentioning that the Elastic Net model performs comparably to the SRFE model, although it does not have the same generalization theory [21].

4.2 Approximation of Friedman functions

In this example, we test the HARFE method on the Friedmann functions, which are used as benchmark examples for certain approximation techniques [7, 8, 35, 37]. The

Table 2 Mean-squared test errors of different methods when approximating Friedman functions

Method	f_1	f_2 ($\times 10^3$)	f_3 ($\times 10^{-3}$)
svm	4.36	18.13	23.15
lm	7.71	36.15	45.42
mnet	9.21	19.61	18.12
rForst	6.02	21.50	22.21
ANOVA	**1.43**	17.21	20.69
SRFE, $q = 2$	2.35	5.15	18.88
HARFE, $q = 2$	**1.52**	**1.31**	**10.90**
HARFE, q	3.01	**1.90**	**13.28**

The values for SRFE and HARFE are obtained by training the model on 100 randomly generated training sets and validating them on 100 randomly generated test sets. We test for different q values. In the last row, $q = 5$ for f_1 and $q = d = 4$ for f_2 and f_3. For the HARFE, $\lambda = 1 \times 10^{-3}, 5 \times 10^{-3}$, and 1×10^{-5} for the functions f_1, f_2, and f_3, respectively. The two best values for every function are highlighted in bold

three Friedman functions are defined as, $f_1 : [0, 1]^{10} \to \mathbb{R}$

$$f_1(x_1, \ldots, x_{10}) = 10 \sin(\pi x_1 x_2) + 20 \left(x_3 - \frac{1}{2} \right)^2 + 10 x_4 + 5 x_5,$$

$f_2 : [0, 1]^4 \to \mathbb{R}$

$$f_2(x_1, x_2, x_3, x_4) = \sqrt{(100 x_1)^2 + \left(x_3 (520\pi x_2 + 40\pi) - \frac{1}{(520\pi x_2 + \pi)(10 x_4 + 1)} \right)^2},$$

and $f_3 : [0, 1]^4 \to \mathbb{R}$

$$f_3(x_1, x_2, x_3, x_4) = \arctan \left(\frac{x_3 (520\pi x_2 + 40\pi) - (520\pi x_2 + 40\pi)^{-1} (10 x_4 + 1)^{-1}}{100 x_1} \right).$$

We follow the setup from [37], where both the training and testing input datasets are randomly generated from the uniform distribution $\mathcal{U}[0, 1]^d$, $m_{train} = 200$, and $m_{\text{test}} = 1000$. In addition, Gaussian noise with zero mean and standard deviations of $\sigma = 1.0, 125.0$, and 0.1, are added to the output data. The MSE of various methods when approximating Friedman functions are displayed in Table 2. We include the results of various methods found in [37] and compare against the results of SRFE [21] and HARFE. For HARFE, the nonzero entries of the random weight vectors $\boldsymbol{\omega}$ and the bias terms are sampled from $\mathcal{U}[-1, 1]$. We use $N = 10^4$ features for f_1 and $N = 2 \times 10^3$ features for f_2 and f_3, $s = 200$, and 50 iterations in total. Our proposed method achieves the smallest errors when approximating Friedman functions f_2 and f_3 even when the value of q is unknown (in that case, we assign $q = d$).

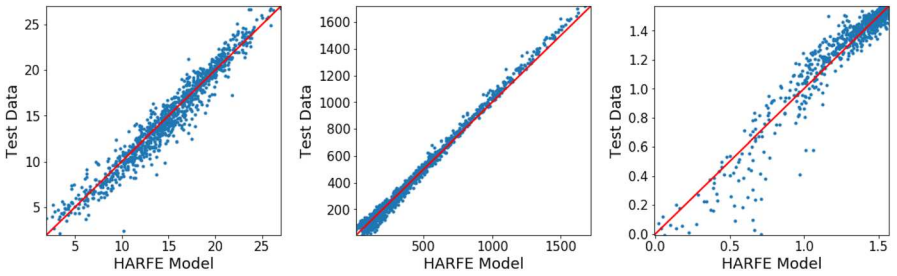

Fig. 2 Scatter plots of the true data versus the predicted values using the HARFE model over the test set for functions f_1, f_2 and f_3 (from left to right)

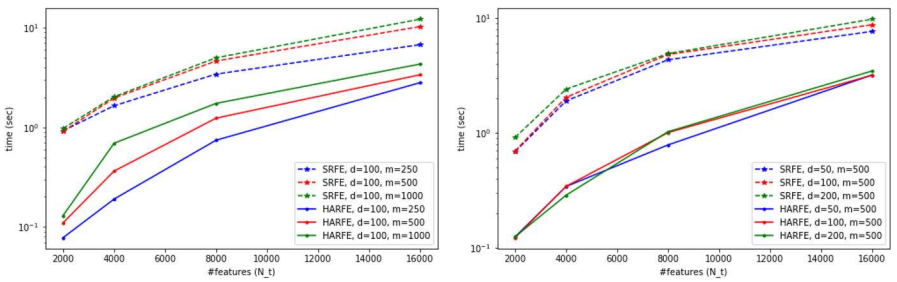

Fig. 3 Plots showing the time required (in seconds) for optimizing the trainable weights $\mathbf{c}$ using HARFE compared with SRFE for $m \in \{250, 500, 1000\}$ with $d = 100$ (left) and $d \in \{50, 100, 200\}$ with $m = 500$ (right) for the function $f(\mathbf{x}) = \sqrt{1 + \|\mathbf{x}\|_2^2}$

It is worth noting that although the first Friedman function has $d = 10$, it is a sparse additive model of order-2, and thus is better approximated by the ANOVA, SRFE, and HARFE models. This is verified by the results in Table 2 with $q = 2$. Note that HARFE with $q = 2$ yields a comparable result with ANOVA for f_1 while outperforming ANOVA in the other two examples. For the second Friedman function, HARFE with $q = 2$ is significantly better than the other methods by almost a factor of 14 times. For the third Friedman function, when the scale of the data is taken into consideration, all methods produce slightly worse results, with HARFE producing the lowest error overall. In Fig. 2, scatter plots show the true data compared to the predicted values using the HARFE model over a test set for functions f_1, f_2 and f_3. The HARFE model produces lower variances for f_1 and f_2, while some bias occurs in f_3 near zero.

In Fig. 3, we plot the runtimes (in seconds) of both HARFE and SRFE during the training phase for $f(\mathbf{x}) = \sqrt{1 + \|\mathbf{x}\|_2^2}$ with different m and d. The stopping criteria used for both the algorithm was same. We can see clearly that HARFE is almost 2.5 times faster than SRFE as the number of features N increases.

In the next experiment, we would like to test HARFE for feature selection. Figure 4displays a histogram illustrating the distribution of indices retained by the HARFE approach for the Friedman function $g : [0, 1]^{20} \to \mathbb{R}$,

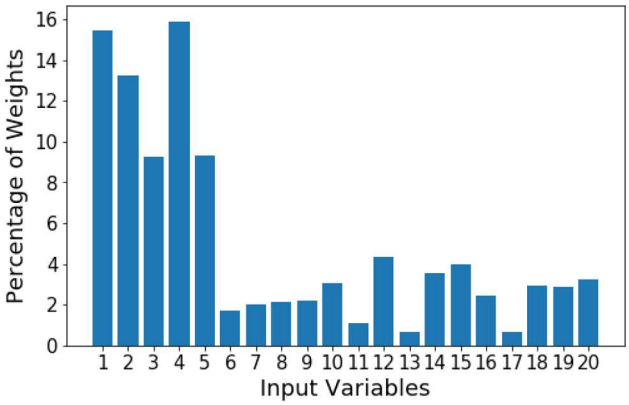

Fig. 4 A histogram plot displaying the distribution of indices retained by the HARFE approach applied to the Friedman function $g(x_1, \ldots, x_{20}) = 10\sin(\pi x_1 x_2) + 20\left(x_3 - \frac{1}{2}\right)^2 + 10x_4 + 5x_5$ using $q = 2$. The histogram is based on the occurrence rate (as a percentage) of the input variables obtained from the HARFE model. The HARFE model correctly identifies the dominate index set

$$g(x_1, \ldots, x_{20}) = 10\sin(\pi x_1 x_2) + 20\left(x_3 - \frac{1}{2}\right)^2 + 10x_4 + 5x_5.$$

In this experiment, we choose $q = 2$. From the histogram in Fig. 4, we observe that HARFE can redistribute the weights based on active input variables especially when applied for a q-order additive function satisfying $q \ll d$ and the number of active variables (five in this case) is much less than the input dimension (which is twenty) of the function. Specifically, the histogram is based on the occurrence rate (as a percentage) of all twenty variables obtained from the HARFE model. The top 5 indices correspond exactly with the correct set.

5 Numerical results on real datasets

We compare the performance of models obtained by the HARFE algorithm with other state-of-the-art sparse additive models [21, 25, 29] when applied to eleven real datasets. An overview of the datasets and the hyperparameters $s, q, m\lambda$ used in the HARFE model are presented in Table 3.

The results of COSSO, Lasso, SALSA, SpAM, and SSAM are obtained from [25, 29, 41] and we include the results of the SRFE and HARFE model. The experiments follow the setup from [25, 29, 41], where the training data is normalized so that the input and output values have zero mean and unit variance along each dimensions. Each dataset is divided in half to form the training and testing sets. The results and comparisons are shown in Table 4. The HARFE approach produces the lowest errors on eight datasets and achieves comparable results on the remaining datasets. Specifically, we significantly outperform other methods on the Propulsion, Airfoil and Forestfire datasets.

Table 3 Overview of eleven datasets and the values of s, $m\lambda$, and q used in the HARFE model

Dataset	Dim	Train	Val	N	s	$m\lambda$	q
Propulsion	15	200	200	3 k	300	10^{-10}	2
Galaxy	20	2000	2000	10 k	1 k	10^{-7}	2
Skillcraft	18	1700	1630	20 k	1 k	1.0	2
Airfoil	41	750	750	80 k	5 k	1.0	2
Forestfire	10	211	167	4220	422	0.5	2
Housing	12	256	250	10 k	1 k	0.1	2
Music	90	1000	1000	10 k	666	2.5	3
Insulin	50	256	250	2560	170	2.75	2
Speech	21	520	520	20 k	1 k	0.1	2
Telemonitor	19	1000	867	15 k	937	0.1	5
CCPP	59	2000	2000	10 k	1 k	0.05	1

The experimental setup and datasets for each test follow from [14, 25, 29, 41]

Table 4 Average MSE on real datasets using various sparse additive models including COSSO, Lasso, SALSA, SpAM, SRFE, SSAM, and HARFE

	HARFE	COSSO	Lasso	SALSA	SpAM	SRFE	SSAM
Propulsion	**0.0000417**	0.00094	0.0248	0.0088	1.1121	0.0154	–
Galaxy	**0.0001024**	0.00153	0.0239	0.00014	0.9542	0.0012	–
Skillcraft	**0.5368**	0.5551	0.6650	0.5470	0.9055	0.8730	0.5432
Airfoil	**0.4492**	0.5178	0.5199	0.5176	0.9623	0.5702	0.4866
Forestfire	**0.2937**	0.3753	0.5193	0.3530	0.9694	0.4067	0.3477
Housing	**0.2636**	1.3097	0.4452	0.2642	0.8165	0.6395	0.3787
Music	**0.6134**	0.7982	0.6349	0.6251	0.7683	1.0454	0.6295
Insulin	**1.0137**	1.1379	1.1103	1.0206	1.2035	1.6456	1.0146
Speech	0.0238	0.3486	0.0730	**0.0224**	0.6600	0.0246	–
Telemonitor	0.0370	5.7192	0.0863	0.0347	0.8643	**0.0336**	0.0689
CCPP	0.0677	0.9684	0.07395	0.0678	**0.0647**	0.07440	0.0694

The lowest error for each dataset is highlighted in bold

In Fig. 5, we plot the histogram of the percentage of weights corresponding to each variable based on the (sparse) coefficient vector approximated using HARFE (from Table 4) for the Propulsion, Housing, Speech, and Telemonitor datasets, respectively. For the Propulsion test in Fig. 5a, we see that the 14th variable (gas turbine compressor decay state coefficient) and the 15th variable (gas turbine decay state coefficient) have the least contribution to the predictor (Lever Position), while the 3rd variable (gas turbine rate of revolutions) is the most relevant variable to the predictor. From the random sampling, the histograms are initiated uniformly (at least in expectation). This experiment shows that the HARFE algorithm will redistribute the weights and identify important variables as a benefit of the sparsity-promoting aspect.

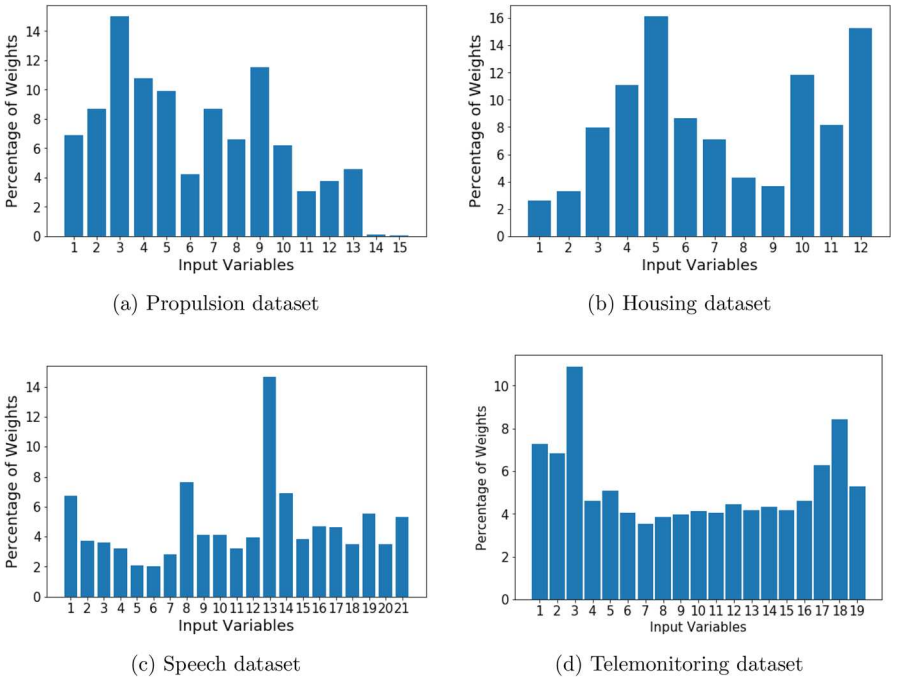

Fig. 5 The histogram plots the percentage of weights corresponding to each variable based on the (sparse) coefficient vector approximated using HARFE for the Propulsion, Housing, Speech, and the Telemonitor datasets, respectively

From Fig. 5b, the HARFE variable selection suggests that the predictor of the House dataset (per capita crime rate by town) is most affected by the 5th variable (proportion of owner-occupied units built prior to 1940) and the 12th variable (median value of owner-occupied homes in $1000's). In Fig. 5c, the plot shows that the 13th (Noise-to-Harmonic or NTH parameter) significantly contributes to the output of the predictor (median pitch) of speech dataset. Lastly, for the Telemonitoring, the experiment in Fig. 5d shows several significant contributors to HARFE trained predictor.

6 Summary

In this work, we proposed a new high-dimensional sparse additive model utilizing the random feature method with two sparsity priors. First, we assumed that the number of terms needed in the model is small which leads to function approximations with low model complexity. Secondly, we enforce a random and sparse connectivity pattern between the hidden layer and the input layer which helps to extract input variable dependencies. Based on the numerical experiments on high-dimensional synthetic examples, the Friedman functions, and real data, the HARFE algorithm was shown to produce robust results that have the added benefit of extracting interpretable variable

information. The analysis of the HARFE algorithm utilizes techniques from compressive sensing and it was shown that the method converges and has a reasonable error bound depending on the number of features, the number of samples, the ridge parameter, the sparsity, the noise level, and dimensional parameters. We expect that risk bounds for the HARFE model can be obtained by following the proofs found in [21]. In ongoing work, we would like to incorporate prior variable dependency information within the construction of the weight matrix.

Acknowledgements E.S. and G.T. were supported in part by NSERC RGPIN 50503-10842. H.S. was supported in part by AFOSR MURI FA9550-21-1-0084, NSF DMS-2331100 and NSF DMS-1752116.

Declarations

Conflict of interest On behalf of all authors, Hayden Schaeffer states that there is no conflict of interest.

Appendices

A Key results

Theorem A.1 *(*Restricted Isometry Constants from [12] *Let the data* $\{\mathbf{x}_k\}_{k\in[m]}$ *be drawn from* $\mathcal{N}(\mathbf{0}, \gamma^2\mathbf{I}_d)$*, the weights* $\{\boldsymbol{\omega}_j\}_{j\in[N]}$ *be drawn from* $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$*, and the random feature matrix* $\mathbf{A} \in \mathbb{C}^{m\times N}$ *be defined component-wise by* $a_{k,j} = \exp(i\langle \mathbf{x}_k, \boldsymbol{\omega}_j\rangle)$*. For* $\eta_1, \eta_2, \delta \in (0,1)$ *and some integer* $s \geq 1$*, if*

$$\begin{aligned} m &\geq C_1\eta_1^{-2}\, s\, \log(\delta^{-1}) \\ \frac{m}{\log(3m)} &\geq C_2\eta_2^{-2}\, s\, \log^2(s)\log\left(\frac{N}{9\log(2m)}+3\right) \\ \sqrt{\delta}\,\eta_1\,(4\gamma^2\sigma^2+1)^{\frac{d}{4}} &\geq N, \end{aligned}$$

where C_1 *and* C_2 *are universal positive constants, then with probability at least* $1-2\delta$*, the* s *restricted isometry constant of* $\frac{1}{\sqrt{m}}\mathbf{A}$ *is bounded by*

$$\delta_s\left(\frac{1}{\sqrt{m}}\mathbf{A}\right) < 3\eta_1 + \eta_2^2 + \sqrt{2}\eta_2.$$

Theorem A.2 (Convergence of HTP Theorem 6.20 from [17]) *Suppose that the* $(6s)^{th}$ *order restricted isometry constant of* $\mathbf{A} \in \mathbb{C}^{m\times N}$ *satisfies* $\delta_{6s} < \frac{1}{\sqrt{3}}$*, then for any* $\mathbf{c} \in \mathbb{C}^N$ *and* $\mathbf{e} \in \mathbb{C}^m$*, the sequence* $\mathbf{c}^k$ *defined by the hard thresholding pursuit with* $y = \mathbf{A}\mathbf{c} + \mathbf{e}$*,* $\mathbf{c}^0 = \boldsymbol{0}$*, using* $2s$ *instead of* s *in the algorithm, satisfies*

$$\|\mathbf{c}^n - \mathbf{c}\|_2 \leq 2\beta^n\|\mathbf{c}\|_2 + \frac{D_1}{\sqrt{s}}\kappa_{1,s}(\mathbf{c}) + D_2\|\mathbf{e}\|_2, \tag{23}$$

for all $n \geq 0$ *where the constants* $\beta \in (0,1)$, $D_1, D_2 > 0$ *depend only on* δ_{6s}. *In particular, if the sequence* $(\mathbf{c}^n)$ *clusters around some* $\mathbf{c}^\sharp \in \mathbb{C}^N$, *then*

$$\|\mathbf{c} - \mathbf{c}^\sharp\|_2 \leq \frac{D_1}{\sqrt{s}} \kappa_{1,s}(\mathbf{c}) + D_2 \|\mathbf{e}\|_2. \tag{24}$$

Lemma A.3 (Lemma 1 in [21]) *Fix the confidence parameter* $\delta > 0$ *and accuracy parameter* $\epsilon > 0$. *Suppose* $f \in \mathcal{F}(\phi, \rho)$ *where* $\phi(\mathbf{x}, \boldsymbol{\omega}) = \exp(i \langle \mathbf{x}, \boldsymbol{\omega} \rangle)$ *and* ρ *is a probability distribution with finite second moment used for sampling the random weights* $\boldsymbol{\omega}$. *Suppose* $N \geq \frac{1}{\epsilon^2}(1 + \sqrt{2 \log(\frac{1}{\delta})})^2$, *then with probability at least* $1 - \delta$, *the following holds with respect to the draw of* $\boldsymbol{\omega}_j$ *for* $j \in [N]$,

$$\|f - f^\star\|_{L^2(d\mu)} \leq \epsilon \|f\|_\rho \tag{25}$$

where $f^\star(\mathbf{x}) = \sum_{i=1}^N \mathbf{c}_k^\star \exp(i \langle \mathbf{x}, \boldsymbol{\omega} \rangle)$, *with* $\mathbf{c}_k^\star = \dfrac{\alpha(\boldsymbol{\omega}_k)}{N \rho(\boldsymbol{\omega}_k)}$.

References

1. Bach, F.: On the equivalence between kernel quadrature rules and random feature expansions. J. Mach. Learn. Res. **18**(1), 714–751 (2017)
2. Bach, F.R.: Consistency of the group lasso and multiple kernel learning. J. Mach. Learn. Res. **9**(6), 20 (2008)
3. Bartlett, P.L., Long, P.M., Lugosi, G., Tsigler, A.: Benign overfitting in linear regression. Proc. Natl. Acad. Sci. **117**(48), 30063–30070 (2020)
4. Belkin, M., Hsu, D., Ma, S., Mandal, S.: Reconciling modern machine-learning practice and the classical bias-variance trade-off. Proc. Natl. Acad. Sci. **116**(32), 15849–15854 (2019)
5. Belkin, M., Rakhlin, A., Tsybakov, A.B.: Does data interpolation contradict statistical optimality? In: The 22nd International Conference on Artificial Intelligence and Statistics, pp. 1611–1619. PMLR (2019)
6. Bertsimas, D., Van Parys, B.: Sparse high-dimensional regression: exact scalable algorithms and phase transitions. Ann. Stat. **48**(1), 300–323 (2020)
7. Beylkin, G., Garcke, J., Mohlenkamp, M.J.: Multivariate regression and machine learning with sums of separable functions. SIAM J. Sci. Comput. **31**(3), 1840–1857 (2009)
8. Binev, P., Dahmen, W., Lamby, P.: Fast high-dimensional approximation with sparse occupancy trees. J. Comput. Appl. Math. **235**(8), 2063–2076 (2011)
9. Blumensath, T., Davies, M.E.: Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. **27**(3), 265–274 (2009)
10. Bouchot, J.-L., Foucart, S., Hitczenko, P.: Hard thresholding pursuit algorithms: number of iterations. Appl. Comput. Harmon. Anal. **41**(2), 412–435 (2016)
11. Campbell, C.: Kernel methods: a survey of current techniques. Neurocomputing **48**(1–4), 63–84 (2002)
12. Chen, Z., Schaeffer, H.: Conditioning of random feature matrices: Double descent and generalization error. arXiv:2110.11477 (arXiv preprint) (2021)
13. Christmann, A., Zhou, D.-X.: Learning rates for the risk of kernel-based quantile regression estimators in additive models. Anal. Appl. **14**(03), 449–477 (2016)
14. Dua, D., Graff, C.: UCI machine learning repository (2017)
15. W. E, C. Ma, Wojtowytsch, S., Wu, L.: Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. arXiv:2009.10713 (arXiv preprint) (2020)
16. Foucart, S.: Hard thresholding pursuit: an algorithm for compressive sensing. SIAM J. Numer. Anal. **49**(6), 2543–2563 (2011)

17. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Birkhäuser, Basel (2013)
18. Frankle, J., Carbin, M.: The lottery ticket hypothesis: finding sparse, trainable neural networks. arXiv:1803.03635 (arXiv preprint) (2018)
19. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011)
20. Harris, K.D.: Additive function approximation in the brain. arXiv:1909.02603 (arXiv preprint) (2019)
21. Hashemi, A., Schaeffer, H., Shi, R., Topcu, U., Tran, G., Ward, R.: Generalization bounds for sparse random feature expansions. Appl. Comput. Harmonic Anal. **62**, 310–330 (2023)
22. Hastie, T., Montanari, A., Rosset, S., Tibshirani, R.J.: Surprises in high-dimensional ridgeless least squares interpolation. Ann. Stat. **50**(2), 949 (2022)
23. Hazimeh, H., Mazumder, R.: Fast best subset selection: coordinate descent and local combinatorial optimization algorithms. Oper. Res. **68**(5), 1517–1537 (2020)
24. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998)
25. Kandasamy, K., Yu, Y.: Additive approximations in high dimensional nonparametric regression via the salsa. In: International Conference on Machine Learning, pp. 69–78. PMLR (2016)
26. Li, Z., Ton, J.-F., Oglic, D., Sejdinovic, D.: Towards a unified analysis of random Fourier features. In: International Conference on Machine Learning, pp. 3905–3914. PMLR (2019)
27. Liang, T., Rakhlin, A.: Just interpolate: kernel "ridgeless" regression can generalize. Ann. Stat. **48**(3), 1329–1347 (2020)
28. Lin, Y., Zhang, H.H.: Component selection and smoothing in multivariate nonparametric regression. Ann. Stat. **34**(5), 2272–2297 (2006)
29. Liu, G., Chen, H., Huang, H.: Sparse shrunk additive models. In: International Conference on Machine Learning, pp. 6194–6204. PMLR (2020)
30. Luedtke, J.: A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. Math. Program. **146**(1), 219–244 (2014)
31. Maleki, A.: Coherence analysis of iterative thresholding algorithms. In: 2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 236–243. IEEE (2009)
32. Mazumder, R., Radchenko, P., Dedieu, A.: Subset selection with shrinkage: sparse linear modeling when the SNR is low. Oper. Res. **71**(1), 129–147 (2023)
33. Mei, S., Misiakiewicz, T., Montanari, A.: Generalization error of random features and kernel methods: hypercontractivity and kernel matrix concentration. Appl. Comput. Harmonic Anal. **59**, 3–84 (2022)
34. Mei, S., Montanari, A.: The generalization error of random features regression: precise asymptotics and the double descent curve. Commun. Pure Appl. Math. **20**, 20 (2019)
35. Meyer, D., Leisch, F., Hornik, K.: The support vector machine under test. Neurocomputing **55**(1–2), 169–186 (2003)
36. Potts, D., Schmischke, M.: Approximation of high-dimensional periodic functions with Fourier-based methods. SIAM J. Numer. Anal. **59**(5), 2393–2429 (2021)
37. Potts, D., Schmischke, M.: Interpretable approximation of high-dimensional data. SIAM J. Math. Data Sci. **3**(4), 1301–1323 (2021)
38. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: NIPS, vol 3, p. 5. Citeseer (2007)
39. Rahimi, A., Recht, B.: Uniform approximation of functions with random bases. In: 2008 46th Annual Allerton Conference on Communication, Control, and Computing, pp. 555–561. IEEE (2008)
40. Rahimi, A., Recht, B.: Weighted sums of random kitchen sinks: replacing minimization with randomization in learning. In: NIPS, pp. 1313–1320. Citeseer (2008)
41. Ravikumar, P., Liu, H., Lafferty, J.D., Wasserman, L.A.: Spam: Sparse additive models. In: NIPS, pp. 1201–1208 (2007)
42. Rudi, A., Rosasco, L.: Generalization properties of learning with random features. In: NIPS, pp. 3215–3225 (2017)
43. Smola, A.J., Schölkopf, B.: Learning with Kernels, vol. 4. Citeseer (1998)
44. Tsigler, A., Bartlett, P.L.: Benign overfitting in ridge regression. Proc. Nat. Acad. Sci. **117**(48), 30063–30070 (2020)
45. Tyagi, H., Kyrillidis, A.Gärtner, B., Krause, A.: Learning sparse additive models with interactions in high dimensions. In: Artificial Intelligence and Statistics, pp. 111–120. PMLR (2016)

46. Xie, W., Deng, X.: Scalable algorithms for the sparse ridge regression. SIAM J. Optim. **30**(4), 3359–3386 (2020)
47. Xie, Y., Shi, B., Schaeffer, H., Ward, R.: Shrimp: Sparser random feature models via iterative magnitude pruning. Math. Sci. Mach. Learn. PMLR **190**, 303–318 (2022)
48. Xu, Z., Jin, R., Yang, H., King, I., Lyu, M. R.: Simple and efficient multiple kernel learning by group lasso. In: ICML (2010)
49. Yen, I.E.-H., Lin, T.-W., Lin, S.-D., Ravikumar, P.K., Dhillon, I.S.: Sparse random feature algorithm as coordinate descent in Hilbert space. In: Advances in Neural Information Processing Systems, pp. 2456–2464 (2014)
50. Zhang, T.: Learning bounds for kernel regression using effective data dimensionality. Neural Comput. **17**(9), 2077–2098 (2005)
51. Zhou, H., Lan, J., Liu, R., Yosinski, J.: Deconstructing lottery tickets: Zeros, signs, and the supermask. Advances In: Advances in neural information processing systems, vol. 32 (2019)