HIERARCHICAL REGULARIZATION NETWORKS FOR SPARSIFICATION BASED LEARNING ON NOISY DATASETS

PRASHANT SHEKHAR AND ABANI PATRA

ABSTRACT. We propose a hierarchical learning strategy aimed at generating sparse representations and associated models for large noisy datasets. The hierarchy follows from approximation spaces identified at successively finer scales. For promoting model generalization at each scale, we also introduce a novel, projection based penalty operator across multiple dimension, using permutation operators for incorporating proximity and ordering information. The paper presents a detailed analysis of approximation properties in the reconstruction Reproducing Kernel Hilbert Spaces (RKHS) with emphasis on optimality and consistency of predictions and behavior of error functionals associated with the produced sparse representations. Results show the performance of the approach as a data reduction and modeling strategy on both synthetic (univariate and multivariate) and real datasets (time series). The sparse model for the test datasets, generated by the presented approach, is also shown to efficiently reconstruct the underlying process and preserve generalizability.

1. INTRODUCTION

Hierarchical learning traditionally involves a sequence of operations based on some hierarchy, for making useful inferences from data. Bayesian hierarchical models for example usually involve a hierarchy of three model classes, the data model, the process model and finally the parameter model [3, 13]. This forms a hierarchy for the updating scheme of the parameters as learning happens sequentially over time. Multiscale models also have an inbuilt hierarchy of approximations, and various research works try to make joint inference on data, by combining these model components in some intelligent fashion [4, 23]. Hierarchical models also have parallels to deep learning models which implement sequential function compositions to learn a data generation mechanism [20, 26].

Motivated by these diverse applications, we present a hierarchical structure of competing regularization networks [19, 39, 38], that make inferences over the observed data. The chosen network has to satisfy the criteria of highest generalizable performance with least model complexity [24]. The requirement of least complexity also allows for generation of a sparse representation for the dataset, making our
approach suitable for data reduction problems \cite{14,18,52}. Our approach introduces a scale parameter $s$ and defines a mapping between $s$ and the corresponding approximation space $\mathcal{H}_s$ in the hierarchy of spaces considered. The main idea of exploiting the inherent correlation structure in the data at multiple levels follows directly from \cite{43}. However, the notion of convergence used in \cite{43} fails if the observations are reported with sampling noise. We have addressed the problem of sparse modeling for such noisy datasets in a similar hierarchical setting.

1.1. Problem setup and definition. Let $Y = (y_1, y_2, ..., y_n) \in \mathbb{R}^n$ are discrete data values observed at $X = (x_1, x_2, ..., x_n) \in \mathbb{R}^{n \times d}$. Considering some true underlying process $f : \mathbb{R}^d \to \mathbb{R}$, the values in $Y$ can be regarded as noisy versions of $f|_X$ ($y_i = f(x_i) + \text{noise}$). We further consider two additional sets. First set $\Omega_x$ contains the data points $x_i$ at which the observations were made ($x_i \in X \subset \Omega_x \subset \mathbb{R}^d$). The observations are samples from a second set, $\Omega_y$ ($y_i \in Y \subset \Omega_y \subset \mathbb{R}$). Now, for a fixed element of $\Omega_x$, we expect a probabilistic distribution on $\Omega_y$. Hence a joint probability distribution $p(x, y)$ can be defined on $\Omega_x \times \Omega_y$. Therefore our training data $D = \{(x_i, y_i) \in \Omega_x \times \Omega_y\}_{i=1}^n$ can be thought of as a result of $n$ samples (i.i.d) from $\Omega_x \times \Omega_y$ according to the distribution $p(x, y)$.

Given such a random noisy data sample $D$, we propose a strategy for data reduction and learning through intelligent sparsification. Data reduction seeks to find a smaller sparse subset $X_s \subseteq X$, that is sufficient for providing acceptable approximations to the underlying process $f|_X$ while also generalizing predictions to unseen data points $x(\in \Omega_x) \notin X$. The learning part is justified by the sparse model produced by the proposed approach, that exclusively uses the subset $X_s$ to make these predictions. Hence in essence, our approach makes the following transformation to the input data

\begin{equation}
\text{(1.1)} \quad \text{full dataset} \Rightarrow \text{sparse representation + sparse model}
\end{equation}

Therefore, the proposed approach can be used to replace large noisy datasets with a smaller subset and an associated model that can be used to make all future predictions. The strategy may also be used to construct effective surrogates of complex computer models by sampling outputs. We note the strategy is provably good for prediction in the domain of observation.

1.2. Proposed solution framework. Given such a problem setup, we are required to learn a function $\hat{f} \in \mathcal{H}$ (native Reproducing Kernel Hilbert Spaces (RKHS)) which is closest (within some measure) to being the underlying process generating observations $Y$ at $X$. For dealing with the ill-posedness of the problem of fitting noisy data, additional smoothness constraints are applied. Thus we have the following variational problem as our objective

\begin{equation}
\text{(1.2)} \quad \hat{f} = \arg \min_{\tilde{f} \in \mathcal{H}} \left[ \frac{1}{n} V(Y, \tilde{f}|_X) + \lambda \cdot \zeta(\tilde{f}) \right]
\end{equation}

Here $V(,\cdot)$ is a loss function and $\zeta(\tilde{f}) = ||Z\tilde{f}||^2_{\mathcal{H}}$ is called a stabilizer, where $Z$ usually is a differential operator and $||\cdot||_{\mathcal{H}}$ is the native RKHS norm. For example, if we make the following choices in 1-dimension

\begin{itemize}
\item[1] The code for the proposed approach is available online \url{https://github.com/prshhekhar-tufts/Hierarchical_noisy.git}
\end{itemize}
\begin{equation}
V(Y, \hat{f}|X) = \sum_{i=1}^{n} (y_i - \hat{f}_i)^2 \quad \text{and} \quad \zeta(\hat{f}) = \|Z\hat{f}\|^2 = \int_{\Omega_{s}} \left[\frac{d^2 \hat{f}(x)}{dx^2}\right] dx
\end{equation}

then the function which minimizes (1.2) is a spline \cite{50, 27}. Also \( \lambda \) in (1.2) is the regularization parameter which maintains a balance between approximation accuracy and smoothness. We obtain the classical \((L_2)\) regularization network if we use squared error loss (as in (1.3)) in formulation (1.2). \cite{39} revealed this relationship between algorithms implementing regularization induced smoothness, with Multilayer Neural Networks.

Work presented here proposes to extend the hierarchical algorithm from \cite{43} to noisy datasets by solving the variational problem (1.2) at multiple scales (equivalent to fitting multiple competing regularization networks) and inferring the network (indexed by scale) that is most appropriately able to model the observations reported. The measure of ‘appropriateness’ will be discussed in more detail in the following sections. Given the random sample of data \( D = \{(x_i, y_i) \in \Omega_x \times \Omega_y\}_{i=1}^{n} \), our approach considers a sequence of scale dependent RKHS \( H_s \), with an associated kernel \( K^s : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R} \), allowing (for each scale \( s \)) us to write a noisy data model of the form

\begin{equation}
Y = \mathcal{T}^s f + \varepsilon
\end{equation}

Here \( \varepsilon \sim N(0, \sigma^2 I) \in \mathbb{R}^n \) is a generic error term at each scale, with function \( f \in H_s \) (assumed) being the true latent process to be inferred. \( \mathcal{T}^s \) is an evaluation functional defined as \( \mathcal{T}^s f = (f_1, f_2, ..., f_n)^T \in \mathbb{R}^n \). As evaluation functionals are bounded and linear in RKHS, therefore \( \mathcal{T}^s \in \mathcal{B}(H_s, \mathbb{R}^n) \). Hence given data \( D \), our approach fits the model of type (1.4) by considering a sequence of scale dependent approximation spaces \( H_s \) (to infer \( f \in H_s \)). More specifics on \( H_s \) are provided in the subsequent sections. The scale \( s \) with the best approximation \( (A_s f : \mathbb{R}^d \rightarrow \mathbb{R} \text{ where } A_s f \in H_s) \) to \( f \) (among the discretized scales considered in the scale space) is then returned as the convergence scale \( t \).

While generating scale dependent models for the data, our hierarchical approach also creates a series of corresponding sparse subsets \( (X_1, X_2, ..., X_s, ...) \) which consist of representative data points from \( X \) \( (X_s \subseteq X) \) \cite{9, 46} chosen intelligently by the algorithm. The cardinality (number of data points) of these subsets follow the relation

\[ |X_1| \leq |X_2| \leq ... \leq |X_s| \leq ..|X| \quad \text{where} \quad |\cdot| \text{ is the cardinality operator} \]

Here, it should be noted that the the approximations \( (A_s f : \mathbb{R}^d \rightarrow \mathbb{R} \text{ at each scale only use the datapoints in the corresponding sparse subset } X_s) \) This enables efficient inference from a reduced version of the original dataset \( D \) and justifies the transformation in (1.1).

The scope of application of the ideas presented in this paper is general in both problems targeted and proposed approach, with relations to many other research problems. For example, multiresolution analysis provides one of the earliest references on multiscale processing of datasets \cite{35, 15}. There is also a rich literature on geometric data analysis with diffusion maps incorporating the ideas of multiscale analysis \cite{11, 12, 34}. The hierarchy in our approximation spaces is closely related to Hierarchical Radial Basis Functions (HRBF) \cite{21, 6}. These research
works focus on combining models at multiple scales to appropriately capture an underlying process. This idea of multiscale basis functions also forms the foundation in more recent works like [4], where the authors project the error orthogonal to the approximation space of previous scales to the next scale. This idea was also explored before by [23]. For our problem, since we are targeting noisy data, instead of combining the scales to reduce fitting error, we consider one scale at a time and incorporate an additional regularization parameter that promotes generalization. Since our approach generates data driven hierarchical basis functions belonging to RKHS, therefore the proposed approach is also related to work such as [10] and [1], where the authors consider data dependent multiscale dictionaries that generalize wavelets in geometric sense. The physics based models have utilized the idea of multilevel modeling through multigrid methods [7, 44]. There are many related papers in the general field of data analysis and machine learning (see for e.g. [25, 32] ) relating the idea to our approach. Since, the current work focuses on generating hierarchical basis functions, it is also closely related to works such as [5, 28] that implement the idea of sparse grids for data analysis and learning tasks.

1.3. Contributions. The principal contributions of this paper can be summarized as follows:

• A hierarchical approach to data reduction and modeling using a sparse representation of the dataset is introduced. This enables us to replace a large noisy datasets with its sparse representation and an associated model for making any future predictions and generalizations.

• The paper also proposes a novel type of smoothing penalty in multiple dimensions based on projections. This is achieved through a set of permutation operators for implementing localized penalties of varying degree.

• The paper also develops and presents theoretical foundations for the approximation and consistency properties of the proposed algorithm. This is followed by a detailed analysis of bounds on approximation operators and error in mean approximations.

2. Hierarchical Learning Approach

In our previous work [43], building on the work in [4], we introduced and developed a methodology of data reduction (for noiseless data) through efficient basis construction exploiting the correlation structure present in the data. This algorithm was based on getting a relevant set of trial functions sampled as columns from a discrete kernel function. The scale at which these basis functions were able to efficiently approximate the observed data in the least square sense was considered as the convergence scale. The approach constructed a sequence of scale (s) dependent approximations (represented as \((A_1f), (A_2f), (A_3f), \ldots, (A_sf), \ldots\)) to the unknown function \(f : \mathbb{R}^d \rightarrow \mathbb{R}\) by considering a hierarchy of approximation spaces \(H_s\). Each of these approximations used a subset of dataset \(X_1, X_2, X_3, \ldots, X_s, \ldots \subseteq X\) respectively for learning. Since the data was noiseless (\(f|_X\) was directly observed instead of \(Y\)), the target function was projected on the sampled basis at each scale by solving the following optimization problem

\[
A_sf = \arg \min_{\tilde{f} \in \Gamma^s} \left[ V(f|_X, \tilde{f}|_X) \right] = \arg \min_{\tilde{f} \in \Gamma^s} \left[ ||f|_X - \tilde{f}|_X||_2^2 \right]
\]
Here $\Gamma^s$ is the subspace defined at each scale $s$ in the native RKHS as
\begin{equation}
\Gamma^s = \text{span}\{K^s(\cdot, x_i) : x_i \in X_s\} \approx \text{span}\{K^s(\cdot, x_j) : x_j \in X\}
\end{equation}
with $K^s(\cdot, \cdot)$ being the reproducing kernel for the RKHS $H_s$.\cite{2,31,3} and formulation (2.1) being the standard problem of orthogonal projection \cite{42}.

In this paper we extend this idea to noisy datasets, where models cannot rely completely on the observations (as they are corrupted with noise). So we ameliorate the effect of noise by introducing a penalty function for inducing smoothness (under the common assumption that noise induces false rapid fluctuations \cite{50}) thus obtaining the following constrained projection formulation (same as the $L_2$ regularization network functional as in (1.2)).

\begin{equation}
A_s f = \arg \min_{f \in \Gamma^s} \left\{ \frac{1}{n} ||Y - \mathcal{T}^s \hat{f}||_2^2 + \lambda_s ||J_s \hat{f}||_{H_s}^2 \right\}
\end{equation}

Here $J_s$ is a suitable projection operator on [$\Gamma^s$] (a particular choice of $\zeta(\cdot)$) which allows efficient penalization (regularization) of sharp changes in $\hat{f}$. $\mathcal{T}^s$ is an evaluation functional defined in (1.4). The solution to (2.3) has a form $A_s f = \sum_{i=1}^{\lfloor |X_s|/s \rfloor} \theta_i K^s(\cdot, x_i)$ (from \cite{37}, $x_i \in X_s$), with $\theta_i$ being suitable basis weights minimizing the cost objective 2.3 and $K^s$ being the reproducing kernel for $H_s$.

### 2.1. Regularization structure

Following standard procedures in kernel based approximation methods \cite{37,50}, it is often desirable to only penalize certain specific functions in $H_s$ and keep the rest of the functions unpenalized (which is achieved precisely by the projection operator $J_s$, in (2.3)). Let $H_{s,0} = \text{span}\{\psi_0, \psi_1, \ldots, \psi_{p_1}\}$ be a subspace of $H_s$ containing these unpenalized functions with its orthogonal complement $H_{s,1} = H_{s,0}^\perp = \text{span}\{\phi_0, \phi_1, \ldots, \phi_{p_2}\}$ spanned by the functions whose behavior needs to be constrained. Therefore $H_s = H_{s,0} \oplus H_{s,1}$ (also $p_1 + p_2 = |X_s|$). Coming back to (2.3), we conclude that a suitable $J_s$ has $H_{s,0}$ as its null space with $H_{s,1}$ being its projection or range space. \cite{2} also showed that $H_{s,0}$ and $H_{s,1}$ are themselves valid RKHS with suitable Kernels $K_0^s$ and $K_1^s$ respectively such that $K^s = K_0^s + K_1^s$. The projection operator $J_s$ can take various forms \cite{27,18,47}, however for our hierarchical approach we have chosen to implement a difference operator based penalty on the projections across each dimension (similar to the one used by \cite{18}). For better understanding of the penalty operator, consider a Relation $R$ (\leq: less-than-or-equal) \cite{20} defined on the domain set $\Omega_x \subset \mathbb{R}$ (univariate approximation) such that $\Omega_\xi$ is partially ordered by $R$. Therefore corresponding to each $x \in \Omega_x$, we can define a function $K^s(\cdot, x)$ and associate a weight $\theta^x$ with it, making weights a function of the continuous variable $x$ ($\theta^x$ is used in the penalty definition in (2.4) and (2.5)). Now considering the discrete case and applying the same ordering $R$ on $\Theta^s_{qr} = \{\theta^x_i | x_i \in X_s\}$, represented as $\Theta^s_{x} = P \epsilon^s \Theta^s_{x}$. Here $P \epsilon^s$ is the permutation operator at scale $s$ in the x-direction (enforcing relation $R$) and $\Theta^s_{x}$ is the set of coordinates for the bases set spanning the approximation space $\Gamma^s$. The initial ordering of $\theta^x_i \in \Theta^s_{x}$ is determined by the ordering of the corresponding basis functions in the bases set. In the current research we implement the penalization of sharp changes by constraining the behavior of basis functions at data points ($x_i$) in close proximity (as per the ordering induced by $R$) to vary in a smooth manner. This is achieved by constraining the rate of change of the weights of these basis functions. Thus for a univariate function $\hat{f} = B^s \Theta^s_{qr}$ (where
$B^s$ is the spanning basis for $\Gamma^s$), we consider the following proxies for the first and second order derivative based penalties.

\begin{equation}
\zeta(\tilde{f})_{q=1} = \int_{\Omega_x} \left[ \frac{d\theta^x}{dx} \right]^2 dx \approx ||D^1\Theta^x_s||^2 = \Theta^x_{qr}^T P e_x^s T D^1 T D^1 P e_x^s \Theta^x_{qr}
\end{equation}

\begin{equation}
\zeta(\tilde{f})_{q=2} = \int_{\Omega_x} \left[ \frac{d^2\theta^x}{dx^2} \right]^2 dx \approx ||D^2\Theta^x_s||^2 = \Theta^x_{qr}^T P e_x^s T D^2 T D^2 P e_x^s \Theta^x_{qr}
\end{equation}

Here $D^q$ is a difference operator of order $q$ on $\Theta^x_s$. Beginning with the difference operator for individual $\theta_i \in \Theta^x_s$ (represented as $\Delta^q$ for $q$th order penalty) we have

$\Delta^1 \theta_i = \theta_i - \theta_{i-1}$

$\Delta^2 \theta_i = \Delta^1 (\Delta^1 \theta_i) = \theta_i - 2\theta_{i-1} + \theta_{i-2}$

$\vdots$

$\Delta^q \theta_i = \Delta^1 (\Delta^{q-1} \theta_i)$

And in matrix form, $\Delta^q$ represented as $D^q$ can be expressed as follows (considering 5 basis functions and $q = 1, 2$ respectively as example)

$D^1 = \begin{pmatrix}
-1 & 1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & -1 & 1
\end{pmatrix}$

$D^2 = \begin{pmatrix}
1 & -2 & 1 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 \\
0 & 0 & 1 & -2 & 1
\end{pmatrix}$

Based on the requirement, it is straightforward to come up difference operators for higher order penalties ($D^q$ for $q > 2$). It should be noted that (2.4) and (2.5) indeed define a seminorm on the space $\mathcal{H}_s$, again confirming the fact that these norms are evaluated in some subspace of $\mathcal{H}_s$ (just penalizing the projection in the subspace $\mathcal{H}_{s,1}$).

Coming back to problem (2.3), the loss function and the stabilizing operator can be represented as

\begin{equation}
V = ||Y - T^s \tilde{f}||_2^2 = ||Y - B^s \Theta^x_{qr}||_2^2
\end{equation}

\begin{equation}
\zeta(\tilde{f})_q = \Theta^x_{qr}^T P e_x^s T D^q T D^q P e_x^s \Theta^x_{qr}
\end{equation}

Now putting (2.6) and (2.7) in (2.3) leads to the following modified formulation for univariate approximations

\begin{equation}
\min_{\Theta^x_{qr} \in \mathbb{R}^{N \times 1}} \left[ \frac{1}{n} ||Y - B^s \Theta^x_{qr}||_2^2 + \lambda \Theta^x_{qr}^T P e_x^s T D^q T D^q P e_x^s \Theta^x_{qr} \right]
\end{equation}

For modeling in higher dimensions, we put independent penalties in each dimension in a similar way as before. Let $\Theta^i$ is the ordering of the weight vector as per the Relation $\preceq$ on coordinates in the $i$th dimension and $P e_i^s$ is the corresponding permutation operator which transforms $\Theta^x_{qr}$ ($\Theta^i = P e_i^s \Theta^x_{qr}$). Also let $Q = [q_1, q_2, \ldots, q_d]$ be the vector of order of penalties across each of the dimensions
Figure 1. Nature of penalty for 2-D basis functions imposed by projection on the corresponding dimensions and application of a permutation operator

For illustrating the penalty structure, we have presented a test case in Figure 1. Here we have the X-Y plane as the approximation domain. Assuming at any scale \( s \), \( \Theta_{s_{qr}} = [\theta_1, \theta_2, \theta_3, \theta_4] \). Depending on the location of these basis function (in the data space), we have the following permutation operators

\[
P_{e_x^s} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P_{e_y^s} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]

giving us \( \Theta_x^s = [\theta_1, \theta_3, \theta_2, \theta_4] \) and \( \Theta_y^s = [\theta_1, \theta_2, \theta_3, \theta_4] \) (coefficients according to the ordering \( R \) (\( \leq \)) as described before).

Hence we have the following analogous problem formulation to (2.3) for the L-2 regularization network in higher dimensions

\[
A_s f = \arg \min_{f \in \mathcal{T}^s} \left[ \frac{1}{n} ||Y - T^s \hat{f}||_2^2 + \sum_{i=s}^{d} \lambda_i ||J_i^s \hat{f}||_{\mathcal{H}_s}^2 \right]
\]

and rewriting it with basis and penalty operators, we obtain the following regularization network problem.
Algorithm 1 Main Algorithm

1: **INPUT:**
   - **Parameters:** \( (T > 0, M > 1) \in \mathbb{R}^2 \)
   - **Dataset:** \( D = \{(x_i, y_i) \in \Omega_x \times \Omega_y \}_{i=1}^n \)
   - **Prediction points:** \( X_m \subset \Omega_x \)

2: **OUTPUT:**
   - **Convergence length scale:** \( \epsilon_t \in \mathbb{R} \)
   - **Sparse model:** \( X_t \subseteq X, C_t \in \mathbb{R}^{|X_t|} \)
   - **Sparse representation:** \( D_t = (X_t, Y_t) \subseteq D \)
   - **Predictions at \( X_m \):** \( P_m \in \mathbb{R}^m, std_m \in \mathbb{R}^m \)

3: Initialize: \( s = 0, l_s = 0, T_h = 0 \)
4: while \( l_s < n \) do
5:   Compute covariance kernel: \( G_s \) on \( X \) with \( \epsilon_s = T/M^s \)
6:   Update numerical rank for current scale: \( l_s = \text{rank}(G_s) \)
7:   Remove sampling bias: \( W = AG_s \) with \( A = [a_{i,j}] \in \mathbb{R}^{k \times n} \) \( a_{i,j} \sim \mathcal{N}(0, 1) \)
8:   Generate permutation information: \( WP_{qr} = QR \)
9:   Produce sparse representation and corresponding bases: \( (X_s, Y_s) \) and \( B^s \)
10: \([\hat{\Lambda}_s, \hat{Q}, \text{Cost}_s] \leftarrow \text{GCV}_\text{model}_\text{evaluate}(B^s, D) \) (illustrated in 2.19)
11: Compute the optimal weights: \( \hat{\Theta}_{qr} \) from (2.13)
12: if \( s = 0 \) or \( \text{Cost}_s < T_h \) :
13: \([t, \epsilon_t, X_t, Y_t, C_t, \Lambda_t, Q_t, T_h] \leftarrow [s, \epsilon_s, X_s, Y_s, \hat{\Theta}_{qr}^s, \hat{\Lambda}_s, \hat{Q}, \text{Cost}_s] \)
14: Update scale: \( s = s + 1 \)
15: end while
16: \( P_m \leftarrow \text{Predict}_\text{mean}(\epsilon_t, X_t, C_t, X_m) \) (Algorithm 2)
17: \( std_m \leftarrow \text{Predict}_\text{CI}(\epsilon_t, X_t, C_t, X_m, \Lambda_t, Q_t, D) \) (Algorithm 3)
18: return \( [\epsilon_t, (X_t, C_t), (X_t, Y_t), P_m, std_m] \)

(2.11) \[
\min_{\Theta_{qr}^s \in \mathbb{R}^{n \times 1}} \left[ \frac{1}{n} \| Y - B^s \Theta_{qr}^s \|_2^2 + \Theta_{qr}^s \top \mathcal{P}_s Q \Theta_{qr}^s \right]
\]

2.2. Fitting the regularization network at multiple scales. The theory of regularization networks has been developed closely in relation to the Vapnik’s ideas on statistical learning theory [49]. If we have a finite set of training data, then the approximation has to be constrained to a small hypothesis space (\( \Gamma^* \)). This concept has been formalized through the capacity of a set and controlling its capacity for proper generalizable approximations. This implementation of capacity control exactly corresponds to finding the optimal \( \Lambda_s \) for a justified trade-off. In this research, we implement and analyze the performance of Generalized Cross-Validation (GCV) for evaluating the performance (quality) of the model at a particular scale. The scale with the minimum optimized GCV metric is regarded as the convergence scale [41] and the corresponding regularization network is declared as the winner and the most suitable for modeling the given dataset \( D \)

Working with the regularization problem (2.11), if we differentiate the cost function with respect to \( \Theta_{qr}^s \), we obtain the normal equations
\[
\frac{1}{n} B^s T B^s + \mathcal{P}^Q_s \right) \hat{\Theta}^s_{qr} = \frac{1}{n} B^s T Y
\]

giving us the \( \hat{\Theta}^s_{qr} \) as a function of hyperparameters \( \Lambda_s = [\lambda^1_s, \lambda^2_s, ..., \lambda^d_s] \)

\[
\hat{\Theta}^s_{qr}(\hat{\Lambda}_s) = \left[ B^s T B^s + n \hat{\mathcal{P}}^Q_s \right]^{-1} B^s T Y
\]

Here \( \hat{\mathcal{P}}^Q_s \) represents the estimated penalty operator \( \mathcal{P}^Q_s \) after substituting optimal hyperparameters \( \Lambda_s \) (represented as \( \hat{\Lambda}_s \)). Therefore, the whole objective of model fitting on the dataset reduces to choosing the right \( \Lambda_s \) (hyperparameters quantifying regularization along each dimension).

Moving forward, we discuss the main algorithm which precisely does this for all the competing, scale dependent regularization networks and chooses the one with the highest generalizable performance. If two scales have the same model fitting cost, then the one with less complexity is chosen (less number of data points in the sparse set \( X_s \)).

Our approach (Algorithm 1), takes a dataset, where a data point \( x_i \in \mathbb{R}^d \) is mapped to an observed value \( y_i \in \mathbb{R} \). In matrix form \( Y = (y_1, y_2, ..., y_n) \) values are obtained at data points \( X = \{x_1, x_2, ..., x_n\} \) \( (Y \in \mathbb{R}^n \) and \( X \in \mathbb{R}^{n \times d} \)). The scalars \( [T, M] \in \mathbb{R}^2 \) are the algorithmic hyperparameters defined by the user. These choices inform the structure of the positive definite function \( (K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}) \) used in the algorithm. Here we work with the squared exponential kernel (2.14) for mapping the covariance structure and generating the space of trial functions \( \Gamma^s \) at each scale \( s \).

\[
G_s(x_i, x_j) = \exp \left( -\frac{||x_i - x_j||^2}{\epsilon_s} \right); \quad \epsilon_s = \frac{T}{M^s}
\]

Here \( \epsilon_s \) is the length scale parameter determining the support of the basis set at scale \( s \). \( M \) is assumed to be 2 (Based on [4]). This choice of \( M \) reduces the length scale of the kernel \( (G_s) \) by a factor of 0.5 at each scale increment, providing an intuitive understanding of how the support of basis functions is adapted to scale variation. Furthermore, if we assume the diameter of the dataset to be distance between the most distant pair of datapoints, then \( T \) is given by

\[
T = 2(\text{Diameter}(X)/2)^2
\]

Besides these parameters, the algorithm also accepts \( X_m = (x_1, x_2, ..., x_m) \subset \Omega_x \subset \mathbb{R}^d \), which represent the data points at which the user wants to predict the underlying function.

In this section we explain how we infer the convergence scale \( (t) \) and the sparse set \( X_t \). The final prediction at the convergence scale will be explained in detail in the following section. Given the Dataset \( D = \{(x_i, y_i) \in \Omega_x \times \Omega_y\}_{i=1}^n \), Algorithm 1 begins with the computation of the covariance operator \( G_s \) (2.14). However, based on research such as [17, 20], the distribution of the dataset might lead to ill-conditioning of this covariance kernel. Therefore we carry out a column pivoted QR decomposition to identify the space \( \Gamma^s \) (at each scale) which approximates the span of the trial functions \( K^s(\cdot, x_j), [1 \leq j \leq n] \) at scale \( s \) (2.2). The QR decomposition
is carried out on \( W \) (instead of \( G \)) directly for obtaining the Permutation matrix \( P_{qr} \). \( W \) is produced by the product of a random normal matrix \( A \) with the \( G \). Here we have \( A \in \mathbb{R}^{k \times n} \) with \( (l_s = \text{rank}(G)) \leq k \leq n \). For our experiments we have assumed \( k = l_s + 8 \) (as in [4]), which means we sample 8 additional rows to account for numerical round-offs during the QR decomposition. The permutation matrix \( P_{qr} \) produced by the decomposition captures the information content of each column of \( W \). \( P_{qr} \) is then used to extract independent columns with the biggest norm contributions (forming the bases set \( B \)) along with the observation points \( (X_s) \) these columns correspond to in the covariance kernel \( G \). The ordering of basis functions in \( B \) (governed by \( P_{qr} \) and representing the information content in decreasing order) determine the ordering of \( \theta \) (here \( x_i \in X_s \)). The dimension of the bases comes from the numerical rank \( (l_s) \) of \( G \) estimated by strategies such as a Rank Revealing QR or a SVD decomposition. Finally \( GCV_{\text{model\, evaluate}} \) subroutine is called which fits the regularization network at the current scale. In essence we follow the ideas from [50] for solving a penalized objective of the form \( (2.11) \), and thus minimize the Generalized Cross Validation metric which is given as

\[
GCV_s(\Lambda_s) = \frac{1}{n} \| (I - U(\Lambda_s))Y \|^2 / \left( \frac{1}{n} \text{Tr}(I - U(\Lambda_s)) \right)^2
\]

where \( U(\Lambda_s) \) is the influence matrix satisfying

\[
T^*(A_s f) = U(\Lambda_s)Y
\]

\[
U(\Lambda_s) = B^* \left[ B^* T B^* + nP_s^Q \right]^{-1} B^* T
\]

Here the objective is to find the optimal penalty vector \( \Lambda_s \). However, besides the regularization parameters \( (\lambda_i) \), we also have to find a suitable penalty order across each dimension \( Q = [q_1, q_2, ..., q_d] \). So, for every dimension \( i \), we just consider \( q_i = 1 \) and 2 (higher order penalties were found to oversmooth approximations weakening the local structure), and choose the final penalty vector \( Q \) (composed of either 1st or 2nd order penalties across each dimension), that lead to a overall smallest \( GCV_s(\Lambda_s) \). Hence, in essence we are solving the following formulation:

\[
\text{Cost}_s = \min_{\Lambda_s > 0, Q_{[q_i \in \{1,2\}}} GCV_s, \quad \text{with} \quad [\hat{\Lambda}_s, \hat{Q}] = \arg \min_{\Lambda_s > 0, Q_{[q_i \in \{1,2\}}} GCV_s
\]

\( GCV_{\text{model\, evaluate}} \) from Algorithm 1 implements this optimization problem. Here \( \Lambda_s > 0 \) refers to \( \lambda_i > 0 \) \forall i.

Therefore, when Algorithm 1 exits the while loop (after covariance kernel becomes numerically full rank), we obtain the convergence scale \( t \) (the scale with the minimum \( \text{Cost}_s \)), the sparse set \( X_t \) and corresponding coordinate of projection \( C_t \) (\( C_t \) is same as \( \Theta_{qr}^t \) at optimal scale \( s = t \) in \( (2.13) \)). Thus, \textbf{we have the sparse representation} \( D_t = (X_t, Y_t) \) \textbf{and the sparse model} \( (X_t, C_t) \) \textbf{for dataset} \( D \).

One additional thing to discuss in Algorithm 1 (before we move on to the \textit{Predict_mean()} and \textit{Predict_CI()} functions in Algorithm 2 and 3 respectively)
is the termination condition for the while loop. For that we provide the following result

**Theorem 1.** The number of while loop iterations for Algorithm 1 are finite and grow with data size $n$ at $O(\log_2(n))$

**Proof.** Following the work of [4], if $\phi$ represents the precision of rank for the Gaussian kernel matrix, then we can define its numerical rank as

$$l_s^\phi(G_s) = \# \left( j : \frac{\sigma_j(G_s)}{\sigma_0(G_s)} \geq \phi \right)$$

where $\sigma_j(G_s)$ is the $j^{th}$ largest singular value of $G_s$. Also if we assume $|V_i|$ represents the length of the bounding box of the data in $i^{th}$ ($i \in [1, d]$) dimension, then given the length scale parameter $\epsilon_s$, the rank of the Gaussian kernel can be bounded above as

$$l_s^\phi(G_s) \leq \prod_{i=1}^{d} \left( \frac{2|V_i|}{\pi} \sqrt{\epsilon_s^{-1} \ln(\phi^{-1}) + 1} \right)$$

Then using proposition 3.7 in [4], we recall the fact that numerical rank of the gaussian kernel matrix is proportional to the volume of the minimum bounding box $Vol = V_1 \times V_2 \times \ldots \times V_d$ and to $\epsilon_s^{-d/2}$. Therefore for a fixed data distribution, following relation holds

$$l_s^\phi(G_s) \propto \epsilon_s^{-d/2} \propto 2^{\alpha d}$$

Hence numerical rank ($l_s$) of $G_s$ increases exponentially with scale $s$ until it becomes full rank ($l_s = n$). The result directly follows from here also establishing the finiteness of the while loop. \square

---

**Algorithm 2** Predict_mean($\epsilon_t, X_t, C_t, X_m$)

1: **INPUT:**
   - Length scale parameter: $\epsilon_t \in \mathbb{R}$
   - Sparse model: $(X_t, C_t)$
   - Prediction points: $X_m \subset \Omega_x$

2: **OUTPUT:**
   - Prediction at $X_m$: $P_m \in \mathbb{R}^m$

3: Compute prediction bases: $B^*_m$ for $X_t$ and $X_m$ (using $\epsilon_t$ [2.14])
4: Compute mean prediction: $P_m = B^*_m C_t$ [2.23]
5: return $P_m$
2.3. Inference at convergence scale ($t$). Algorithm 1 defined the steps for obtaining the convergence scale $t$, the sparse subset $X_t$ and corresponding coordinate of projection $C_t$ (within While loop) for modeling the dataset $D$. However, given a proper approximation space ($\Gamma^t$ spanned by bases centered at the sparse set $X_t$), the second step in modeling is always to generalize this inference over the entire domain. Hence, we use the obtained sparse model $(X_t,C_t)$ to make inference at new data points of interest (Algorithm 2 and 3).

**Algorithm 3** $\text{Predict}\_\text{CI}(\epsilon_t,X_t,C_t,X_m,\Lambda_t,Q_t,D)$

1: **INPUT:**
   - Length scale parameter: $\epsilon_t \in \mathbb{R}$
   - Sparse model: $(X_t,C_t)$
   - Prediction points: $X_m \subset \Omega_x$
   - Hyperparameters: $\Lambda_t \in \mathbb{R}^d, Q_t \in \mathbb{R}^d$
   - Data: $D$

2: **OUTPUT:**
   - Confidence Interval for prediction at $X_m$: $\text{std}_m \in \mathbb{R}^m$

3: Compute data bases: $B^t$ for $X_t$ and $X$ (using $\epsilon_t$ (2.14))
4: Compute $U(\Lambda_t)$: from (2.18) using $B^t$, $\Lambda_t$ and $Q_t$
5: Compute $T^t(A_t f)$: $B^t C_t$
6: Compute $\hat{\sigma}_t^2$: substitute $Y, T^t(A_t f), U(\Lambda_t)$ in (2.24)
7: Compute prediction bases: $B^t_m$ for $X_t$ and $X_m$ (using $\epsilon_t$)
8: Compute the interval ($\text{std}_m$): substitute computed quantities in (2.25)
9: **return** $\text{std}_m$

Starting with the procedure for getting predictions at data points $X_m$ defined in $\text{Predict}\_\text{mean}()$ - shown as Algorithm 2 we formulate the set of bases centered at the sparse set $X_t$ with respect to the prediction location $X_m$ (represented as $B^t_m$), giving the following representation for approximation of the underlying process $f$ restricted to the set $X_m$

$$P_m = A_t f|_{X_m} = B^t_m C_t$$

where $C_t$ (referred to as coordinate of projection) is obtained from Algorithm 1. It is crucial to note here, that for producing these approximations, we just needed the sparse model - $(X_t,C_t)$. We don’t need access to the full dataset $D$. This characteristic of the approach can lead to massive storage and computational savings.

Again, following the ideas of [50], we have presented the steps for getting the confidence intervals (CI) in Algorithm 3 ($\text{Predict}\_\text{CI}()$). Here we use an empirical unbiased estimate of $\sigma_t^2$ for these confidence bounds.

$$\hat{\sigma}_t^2 = \frac{||Y - T^t(A_t f)||^2_2}{df_{\text{res}}}$$

Here $df_{\text{res}}$ represents the degree of freedom for the residual for which we use the non-parametric estimate $df_{\text{res}} = n - 2 \cdot tr(U(\Lambda_t)) + tr(U(\Lambda_t) \cdot U(\Lambda_t)^T)$ (with $U(\Lambda_t)$ as defined in (2.18) at $s = t$). Here $tr$ is the trace operator. Following
the recommendation of [31], the standard deviation for the error term could be estimated as

\[ \text{std}_m((A_t f)(x^*) - f(x^*)) = \sigma \sqrt{B^T B + nP^T Q^{-1} P} \]

Now, it is straightforward to state that 100(1 - \alpha_c)% confidence intervals will be written as

\[ A_t f(x^*) \pm t \left( 1 - \frac{\alpha_c}{2}, df_{res} \right) \text{std}_m((A_t f)(x^*) - f(x^*)) \]

Unlike mean approximation \( A_t f|_{X_m} \), unfortunately, if we want to augment our predictions at new data points \( X_m \) with confidence bounds, then we need to go back to the full dataset \( D \). This is because in (2.25), we need to compute \( B^t \in \mathbb{R}^{|X| \times |X|} \) which involves full data \( X \).

3. Approximation Properties

For developing the results in this section, we have taken ideas from [31, 30, 22, 16]. Here many of the proofs developed consider \( Y \in \text{Dom}(\mathcal{T}) \) with \( Y \in \text{Ker}(\mathcal{T}) \) as a special case. Our first main result provides an inner product representation for the approximation \( A_t f \) to \( f \), produced at scale \( s \). This alternate representation will help us with a more precise consistency and error analysis. Defining \( \delta_x \) as the evaluational functional for \( f \), i.e. \( \delta_x(f) = f(x) \) gives us the dual space \( \mathcal{H}_s = \left\{ \sum_{x \in X} c^s_x \delta^s_x \right\} \), and by assuming the traditional definition of norm in this dual space, we have

\[ K^s(x, y) = \langle K^s(x, \cdot), K^s(y, \cdot) \rangle_{\mathcal{H}_s} = \langle \delta^s_x, \delta^s_y \rangle_{\mathcal{H}_s}, \quad x, y \in \Omega \]

**Definition 1.** Let the pointwise error functional at any data point \( x \in \Omega_x \) has a representation

\[ E^s_{\Lambda_x} = \delta^s_x - M^T_{\Lambda_x}(x) \delta^s_Y \]

Here \( \Lambda_x = [\lambda^1_x, ..., \lambda^d_x] \) is the optimal set of regularization parameters in \( d \)-dimensions. \( \delta^s_x \) is identified as the Riesz representation of the evaluation functional at \( x \) in the dual space of \( \mathcal{H}_s \) and \( M_{\Lambda_x} \) represented as \( M_{\Lambda_x}(x) = [M^1_{\Lambda_x}(x), M^2_{\Lambda_x}(x), ..., M^d_{\Lambda_x}(x)] \in \mathbb{R}^n \), is a set of \( n \) appropriate functions \( \{M^i_{\Lambda_x}\} \) evaluated at \( x \in X \).

Then, given such a representation, we denote the magnitude of expected pointwise approximation error as

\[ \text{Error}(x) = |f(x) - \mathbb{E}[A_t f](x)| = |\delta^s_x(f) - M^T_{\Lambda_x}(x) \delta^s_Y(f)| = |E^s_{\Lambda_x}(f)| \]

Hence from this definition, with some appropriate set of \( n \) functions \( \{M^i_{\Lambda_x}\} \), evaluated at \( x \in \Omega_x \), we have \( \mathbb{E}[A_t f](x) = M^T_{\Lambda_x}(x) \delta^s_Y(f) \).

Moving further, we again define a semi-norm which relates the penalty in multiple dimensions (denoted by \( P^Q_s \)) to the behavior of the basis functions \( B^s \) spanning the approximation space. In essence, this formalizes constraining of the approximation space to limit its capacity.
Definition 2. For bases $B^s$ at scale $s$, we define a semi-inner product and the corresponding semi-norm in $n$-dimensional Euclidean space as

$$\langle a, b \rangle_{T_s} = a^T T_s b \quad ||a||_{T_s} = \langle a, a \rangle_{T_s}^{1/2}, \quad a, b \in \mathbb{R}^n$$

where $T_s$ is a self-adjoint operator satisfying the relation

$$(3.5) \quad P_{qr}|_{X_s} T_s B^s = nP_s^Q$$

Here $P_{qr}|_{X_s} = [I_{|X_s|} \ | 0] \ P_{qr}$, with $P_{qr}$ being the permutation operator for column pivoted QR in Algorithm [1] and $I_{|X_s|}$ is a $|X_s|$-dimensional identity matrix. $P_s^Q$ is the total penalty operator in multiple dimensions.

Now, with the representation of error functional as in (3.2), we state the following result.

Theorem 2. The solution $\hat{M}_{\Lambda_s}(x)$ to the penalized error minimization problem

$$(3.6) \quad \hat{M}_{\Lambda_s}(x) = \arg \min_{M_{\Lambda_s}(x) \in \mathbb{R}^n} \left[ ||E_{\Lambda_s}^x||_{H_s^*}^2 + ||M_{\Lambda_s}(x)||_{H_s^*}^2 \right]$$

satisfies the inner product representations $\langle Y, \hat{M}_{\Lambda_s}(x) \rangle = \langle A_s f \rangle(x)$ for $A_s f$ and

$$(3.8) \quad R_s(x) = G_s M_{\Lambda_s} + T_s M_{\Lambda_s}(x)$$

Proof. Starting with the error functional norm

$$||E_{\Lambda_s}^x||_{H_s^*}^2 = <\delta_s^x - M_{\Lambda_s}^T(x)\delta_s^x, \delta_s^x - M_{\Lambda_s}^T(x)\delta_s^x>_{H_s^*}$$

$$= ||\delta_s^x||_{H_s^*}^2 - 2M_{\Lambda_s}^T(x)\delta_s^x \delta_s^x + M_{\Lambda_s}^T(x)\delta_s^x \delta_s^x T M_{\Lambda_s}(x)$$

Therefore the quantity to be minimized from (3.6) can be written as

$$(3.7) \quad ||\delta_s^x||_{H_s^*}^2 - 2M_{\Lambda_s}^T(x)\delta_s^x \delta_s^x + M_{\Lambda_s}^T(x)\delta_s^x \delta_s^x T M_{\Lambda_s}(x)$$

Now, based on the property of dual space, we know at scale $s$,

$$<\delta_s^x, \delta_s^x>_{H_s^*} = K^*(a, b)$$

Also, let $R_s(x) = \delta_s^x \delta_s^x = (K^*(x, x_1), K^*(x, x_2), ..., K^*(x, x_n)) \in \mathbb{R}^n$ and $G_s = \delta_s^x \delta_s^x T$. Now, differentiating (3.7) with respect to $M_{\Lambda_s}(x)$ and setting it to 0 gives

$$(3.8) \quad R_s(x) = G_s M_{\Lambda_s} + T_s M_{\Lambda_s}(x)$$

Now, since $G_s$ has a rank of $l_s$ at scale $s$ which is also true for orthogonal projection operator for $B^s$ (given as $B^s(B^T B^s)^{-1} B^T$). Therefore in order to sample independent equations from the system (3.8), we use the same method as in Algorithm [1] We again create the matrix $W (= AG_s)$ and carry out a column pivoted QR decomposition $WP_{qr} = QR$. Now applying the permutation operator $P_{qr}$ on system (3.8) and sampling the first $l_s$ equation.

$$P_{qr} R_s(x) = P_{qr} G_s M_{\Lambda_s} + P_{qr} T_s M_{\Lambda_s}(x)$$

For sampling first $|X_s|$ (the cardinality of the sparse set $X_s$ is $l_s$) equations and to remove redundancy, pre-multiplying by $[I_{|X_s|} \ | 0]$ we get

$$R_s(x)|_{X_s} = B^s M_{\Lambda_s} + [I_{|X_s|} \ | 0] P_{qr} T_s M_{\Lambda_s}(x)$$

Using the relation from (3.5)

$$[I_{|X_s|} \ | 0] P_{qr} T_s B^s = nP_s^Q (B^s B^s)^{-1} B^T B^s$$
Therefore, \( B \) hence, the solution to the penalized objective (3.6) in the limit Corollary 2.1. M

\[
\lim_{s \to 0} \Lambda_s = (B^s)^{-1} R_s(x) \in \mathcal{X}
\]

Defining \( \lambda_s \) tends to the unconstrained solution in the same RKHS, establishing the consistency of our constraining procedure. Consistency analysis.]

The proof directly follows from Theorem 2 and using (3.9) as the proof is concluded. □

3.1. Consistency analysis. In this section, we study the behavior of the of the problem formulation 2.10 when we relax the smoothness constraining enforced by the difference based penalty. The results in this section show that as we make the constraints inactive in our penalized formulation, the produced approximation tends to the unconstrained solution in the same RKHS, establishing the consistency of our constraining procedure.

Definition 3. Defining \( \lambda_s^\infty \in \mathbb{R} \) as an upper bound to the set \( \Lambda_s \) (other than the least upper bound) such that

\[
\lim_{\lambda_s^\infty \to 0} (\lambda_s^i/\lambda_s^\infty) \to 0 \quad \forall i \in [1, d] \cap \mathbb{N}
\]

Now, we will provide a corollary (to Theorem 2) explaining the behavior of \( \hat{M}_\Lambda_s(x) \) as \( \lambda_s^\infty \) tends to 0.

Corollary 2.1. The solution to the penalized objective (3.6) in the limit \( \lambda_s^\infty \to 0 \) is the orthogonal projection on the approximation space defined by \( B^s \). Thus on solving

\[
\hat{M}_0(x) = \lim_{\lambda_s^\infty \to 0} \hat{M}_\Lambda_s(x) = \lim_{\lambda_s^\infty \to 0} \left( \arg \min_{\Lambda_s(x) \in \mathbb{R}^n} \left[ ||E_{\Lambda_s}||^2_{H_2} + ||M_{\Lambda_s(x)}||^2_{T_s} \right] \right)
\]

we get \( \hat{M}_0(x) = B^s(B^{s^T}B^s)^{-1} R_s(x) \in \mathcal{X} \) satisfying \( Y, \hat{M}_0(x) = (A_s f)_0(x) \) .

Proof. The proof directly follows from Theorem 2 and using (3.9) as \( \lambda_s^\infty \to 0 \). □

In Corollary 2.1, we have mentioned the approximation \( (A_s f)_0 \), that is obtained by orthogonally projecting on \( B^s \). Hence \( (A_s f)_0(x) = B^s(x)(B^{s^T}B^s)^{-1} B^{s^T} Y \). Next, we provide a theorem relating \( (A_s f)_0 \) to \( A_s f \). This result provides an understanding of the behavior of the produced approximation as constraints become active. However, before getting to the main results we start with a lemma. This lemma provides a tractable representation of inner product of the optimal approximation \( (A_s f) \) at scale \( s \) with any other function \( \tilde{f} \) in the same space (note that \( A_s f, \tilde{f} \in \Gamma^s \)).
Lemma 1. The weighted sum of inner products of projection components for
\( A_s f, \tilde{f} \in \Gamma^s \) along each penalized dimension, admits the Euclidean inner product representation

\[
\sum_{i=1}^{d} \lambda_i^s \left< J_s^i(A_s f), J_s^i(\tilde{f}) \right>_{H_s} = (1/n) \left< Y - T^s(A_s f), T^s \tilde{f} \right>
\]

Proof. We begin our proof by defining a semi-inner product \( \left< \cdot, \cdot \right>_{\Lambda_s} \) on \( \mathbb{R}^n \times \mathbb{R}^n \)

\[
\left< (U_1, U_2), (V_1, V_2) \right>_{\Lambda_s} = \frac{1}{n} \left< U_1, V_1 \right> + \sum_{i=1}^{d} \lambda_i^s \left< J_s^i(T^s U_2), J_s^i(T^s V_2) \right>_{H_s}
\]

Here \( U_1, U_2, V_1, V_2 \in \mathbb{R}^n \). For it to be a valid norm we also assume \( U_2, V_2 \in \text{Dom}(T^s) \) at scale \( s \). Correspondingly we also obtain the semi-inner product induced semi-norm \( || \cdot ||_{\Lambda_s} \) on \( \mathbb{R}^n \times \mathbb{R}^n \)

\[
|| (U, V) ||_{\Lambda_s}^2 = \frac{1}{n} || U ||^2 + \sum_{i=1}^{d} \lambda_i^s || J_s^i(T^s V) ||_{H_s}^2
\]

Now, it can be easily seen that the solution of the Regularized Network at scale \( s \) (problem (2.10)) can be expressed in \( || \cdot ||_{\Lambda_s} \) as follows.

\[
|| (Y, 0) - (T^s(A_s f), T^s(A_s f)) ||_{\Lambda_s}^2 = \inf_{\tilde{f} \in \Gamma^s} || (Y, 0) - (T^s \tilde{f}, T^s \tilde{f}) ||_{\Lambda_s}^2
\]

Therefore, since \((Y, 0) - (T^s(A_s f), T^s(A_s f))\) would be orthogonal to all \((T^s \tilde{f}, T^s \tilde{f}) \in \mathbb{R}^n \times \mathbb{R}^n\) by the property of projections in finite dimensional spaces. Therefore,

\[
\left< (Y, 0) - (T^s(A_s f), T^s(A_s f)), (T^s \tilde{f}, T^s \tilde{f}) \right>_{\Lambda_s} = 0 \ \forall \tilde{f} \in \Gamma^s
\]

\[
\Rightarrow \frac{1}{n} \left< Y - T^s(A_s f), T^s \tilde{f} \right> - \sum_{i=1}^{d} \lambda_i^s \left< J_s^i(A_s f), J_s^i(\tilde{f}) \right>_{H_s} = 0 \text{ using (3.12)}
\]

Thus, the result follows \( \square \)

Coming back to the relation of \( A_s f \) and \( (A_s f)_0 \), we now have the following first result

**Theorem 3.** For any \( \Lambda_s = [\lambda_1^s, \lambda_2^s, \ldots, \lambda_d^s] > 0 \in \mathbb{R}^d \), solution \( A_s f \) to problem (2.10) satisfies

- Pythagoras Theorem

\[
|| Y - T^s(A_s f) ||_{2}^2 = || Y - T^s(A_s f)_0 ||_{2}^2 + || T^s(A_s f)_0 - T^s(A_s f) ||_{2}^2
\]

- Best approximation, if \((A_s f)_0|_X\) is observed instead of \( Y \). Modifying (2.10)

\[
A_s f = \arg \min_{\tilde{f} \in \Gamma^s} \left[ \frac{1}{n} || T^s(A_s f)_0 - T^s \tilde{f} ||_{2}^2 + \sum_{i=1}^{d} \lambda_i^s || J_s^i(\tilde{f}) ||_{H_s}^2 \right]
\]
Proof. (a): substituting $A_s = 0$ in Lemma $1$ we get
\[
\left\langle Y - T^*(A_s f)_0, T^*\hat{f} \right\rangle = 0
\]
Using this,
\[
\|Y - T^*\hat{f}\|_2^2 = \|Y - T^*(A_s f)_0 + T^*(A_s f)_0 - T^*\hat{f}\|_2^2
\]
\[
= \|Y - T^*(A_s f)_0\|_2^2 + 2\left\langle Y - T^*(A_s f)_0, T^*(A_s f)_0 - T^*\hat{f} \right\rangle
\]
\[
+ \|T^*(A_s f)_0 - T^*\hat{f}\|_2^2
\]
\[
= \|Y - T^*(A_s f)_0\|_2^2 + \|T^*(A_s f)_0 - T^*\hat{f}\|_2^2
\]
Replacing $T^*\hat{f}$ by $T^*(A_s f)$ completes the proof
(b): For proving the approximation property, we subtract $\left\langle Y - T^*(A_s f)_0, T^*\hat{f} \right\rangle = 0$ from (3.11), we get
\[
\frac{1}{n} \left\langle Y - T^*(A_s f), T^*\hat{f} \right\rangle - \frac{1}{n} \left\langle Y - T^*(A_s f)_0, T^*\hat{f} \right\rangle = \sum_{i=1}^{d}\lambda_i \left\langle J_i^*(A_s f), J_i^*(\hat{f}) \right\rangle_{\mathcal{H}_s}
\]
Therefore, following Lemma $1$, $A_s f$ is again an optimal solution for the case when $T^*(A_s f)_0$ was observed instead of $Y$.

Again using the following result from Lemma $1$.
\[
(3.15) \quad \sum_{i=1}^{d}\lambda_i \left\langle J_i^*(A_s f), J_i^*(\hat{f}) \right\rangle_{\mathcal{H}_s} = (1/n) \left\langle Y - T^*(A_s f), T^*\hat{f} \right\rangle
\]
We now state our second main result that quantifies the rate of convergence of approximation $A_s f$ to $(A_s f)_0$ and $T^*(A_s f)$ to $T^*(A_s f)_0$ in $\mathcal{H}_s$ and $\mathbb{R}^n$ respectively as $\lambda_s^\infty \to 0$, according to the following convergence order ($g$ is some finite positive constant).
\[
\lim_{\lambda_s^\infty \to 0} \|A_s f - (A_s f)_0\|_{\mathcal{H}_s}^2 \leq \lim_{\lambda_s^\infty \to 0} ng^2\lambda_s^\infty \sum_{i=1}^{d} ||J_s(T^*Y)||_{\mathcal{H}_s}^2 \to 0 : O(\lambda_s^\infty) \text{ in } \mathcal{H}_s
\]
\[
\lim_{\lambda_s^\infty \to 0} \frac{2}{\lambda_s^\infty} ||T^*(A_s f) - T^*(A_s f)_0||_2 \to 0 : o(\lambda_s^\infty) \text{ in } \mathbb{R}^n
\]
Proof. For any function $\hat{f} \in \Gamma^s$, we define a norm as $||\hat{f}||_{\Gamma^s} = ||T^*\hat{f}||_2$. Since $\Gamma^s$ is finite dimensional, therefore norm $|| \cdot ||_{\Gamma^s}$ and $|| \cdot ||_{\mathcal{H}_s}$ would be equivalent on $\Gamma^s$. Thus there would be a constant $g$ ($> 0$) such that
\[
(3.16) \quad ||\hat{f}||_{\mathcal{H}_s} \leq g||\hat{f}||_{\Gamma^s}
\]
Using the result from Lemma $1$ and substituting $\hat{f} = T^{*\dagger}Y - A_s f$
\[
(3.17) \quad \sum_{i=1}^{d}\lambda_i \left\langle J_i^*(A_s f), J_i^*(T^{*\dagger}Y) \right\rangle_{\mathcal{H}_s} - \sum_{i=1}^{d}\lambda_i \left\langle J_i^*(A_s f), J_i^*(A_s f) \right\rangle_{\mathcal{H}_s} = (1/n) \left\langle Y - T^*(A_s f), Y - T^*(A_s f) \right\rangle
\]
On rearranging, we get

\begin{equation}
(3.18) \quad \frac{1}{n}||Y - T^s(A_s f)||^2_{\mathcal{H}_s} + \sum_{i=1}^{d} \lambda_i \|J_i^s(A_s f)\|_{\mathcal{H}_s}^2 = \sum_{i=1}^{d} \lambda_i \langle J_i^s(A_s f), J_i^s(T^s Y) \rangle_{\mathcal{H}_s} \leq \sum_{i=1}^{d} \lambda_i \|J_i^s(A_s f)\|_{\mathcal{H}_s} \|J_i^s(T^s Y)\|_{\mathcal{H}_s}
\end{equation}

Which directly leads to the inequality

\begin{equation}
(3.19) \quad \frac{1}{n}||Y - T^s(A_s f)||^2_{\mathcal{H}_s} \leq \sum_{i=1}^{d} \lambda_i \|J_i^s(T^s Y)\|_{\mathcal{H}_s} \leq \lambda_\infty \sum_{i=1}^{d} \|J_i^s(T^s Y)\|_{\mathcal{H}_s}^2
\end{equation}

Also, putting \( \tilde{f} = A_s f - (A_s f)_0 \) in (3.16) and using (3.13), we additionally get

\begin{equation}
(3.20) \quad ||A_s f - (A_s f)_0||^2_{\mathcal{H}_s} \leq g^2 ||T^s(A_s f) - T^s(A_s f)_0||^2_{\mathcal{H}_s} \leq g^2 ||Y - T^s(A_s f)||^2_{\mathcal{H}_s}
\end{equation}

Using (3.19) and (3.20), the first statement of the theorem follows

\[ ||A_s f - (A_s f)_0||^2_{\mathcal{H}_s} \leq ng^2 \lambda_\infty \sum_{i=1}^{d} \|J_i^s(T^s Y)\|_{\mathcal{H}_s}^2 \]

For the second result, we begin with

\[ \sum_{i=1}^{d} \lambda_i \|J_i^s(A_s f - T^s Y)\|_{\mathcal{H}_s}^2 = \sum_{i=1}^{d} \lambda_i \|J_i^s(A_s f)\|_{\mathcal{H}_s}^2 + \sum_{i=1}^{d} \lambda_i \|J_i^s(T^s Y)\|_{\mathcal{H}_s}^2 - 2 \sum_{i=1}^{d} \lambda_i \langle J_i^s(A_s f), J_i^s(T^s Y) \rangle_{\mathcal{H}_s} \]

On rearranging and using (3.18), we get

\[ \sum_{i=1}^{d} \lambda_i \|J_i^s(A_s f - T^s Y)\|_{\mathcal{H}_s}^2 = \sum_{i=1}^{d} \lambda_i \|J_i^s(A_s f)\|_{\mathcal{H}_s}^2 + \sum_{i=1}^{d} \lambda_i \|J_i^s(T^s Y)\|_{\mathcal{H}_s}^2 - 2 \left[ \frac{1}{n}||Y - T^s(A_s f)||^2_{\mathcal{H}_s} + \sum_{i=1}^{d} \lambda_i \|J_i^s(A_s f)\|_{\mathcal{H}_s}^2 \right] \]

On further solving and normalizing by \( \lambda_\infty \) we get

\[ \frac{2}{n \lambda_\infty} ||Y - T^s(A_s f)||^2_{\mathcal{H}_s} = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_\infty} \|J_i^s(T^s Y)\|_{\mathcal{H}_s}^2 - \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_\infty} \|J_i^s(A_s f - T^s Y)\|_{\mathcal{H}_s}^2 - \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_\infty} \|J_i^s(A_s f)\|_{\mathcal{H}_s}^2 \]

Using (3.13)

\[ \frac{2}{n \lambda_\infty} ||T^s(A_s f) - T^s(A_s f)_0||^2_{\mathcal{H}_s} \leq \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_\infty} \|J_i^s(T^s Y)\|_{\mathcal{H}_s}^2 - \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_\infty} \|J_i^s(A_s f - T^s Y)\|_{\mathcal{H}_s}^2 - \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_\infty} \|J_i^s(A_s f)\|_{\mathcal{H}_s}^2 \]
Now, looking at R.H.S of equation above and using (3.16)

\[ ||J_s^i(A_s f - \mathcal{T}^s Y)||_{\mathcal{H}_s} \leq ||J_s^i||_{\mathcal{H}_s} ||\mathcal{T}^s Y - A_s f||_{\mathcal{H}_s} \leq g||J_s^i||_{\mathcal{H}_s} ||Y - \mathcal{T}^s (A_s f)||_2 \]

Thus with \( \lambda_s^\infty \rightarrow 0 \) from part previous result \( ||J_s^i(A_s f - \mathcal{T}^s Y)||_{\mathcal{H}_s} \rightarrow 0 \). Also

\[ ||J_s^i(\mathcal{T}^s Y)||_{\mathcal{H}_s} - ||J_s^i(A_s f)||_{\mathcal{H}_s} \leq ||J_s^i(A_s f) - J_s^i(\mathcal{T}^s Y)||_{\mathcal{H}_s} \]

Now, since \( \lambda_s^\infty \) tends to 0 faster than \( \lambda_s^\infty \), therefore we conclude

\[ \frac{2}{n \lambda_s^\infty} ||\mathcal{T}^s (A_s f) - \mathcal{T}^s (A_s f)_0||_2^2 \rightarrow 0 \text{ as } \lambda_s^\infty \rightarrow 0 \]

Hence the proof follows. \( \square \)

3.2. **Bounding the approximation behavior.** In this section we analyze the behavior of the approximation produced at individual scales. We provide three results consisting of bounds on the (i) scale dependent approximation operator \( A_s \) (ii) scale dependent approximation at a point \( A_s f(x) \) (iii) scale dependent mean approximation error at a point \( Error(x) = |f(x) - E[A_s f](x)| \). The goal is to show that our formulation behaves in a stable manner as the model is trained to learn from data.

The starting result provides a bound for the approximation at any scale \( s \) with respect to the \( L_\infty \) topology for a compact domain \( \Omega \in \mathbb{R}^d \)

**Theorem 5.** The approximation \( A_s f \) has a \( L_\infty \) upper bound

\[ (3.21) \quad ||A_s f||_{L_\infty} \leq P^s_\infty ||Y||_\infty \]

with \( P^s_\infty \) following the bounds

\[ (3.22) \quad ||U(\Lambda_s)M_0||_2 \leq P^s_\infty \leq ||U(\Lambda_s)M_0||_1 \]

where \( U(\Lambda_s) \) is defined in (2.18) and \( M_0 \) is from corollary 2.1

**Proof.** We begin with the definition of approximation \( A_s f \) expressed as an inner product as in Theorem 2

\[ ||A_s f||_{L_\infty} = \max_{x \in \Omega} |A_s f(x)| = \max_{x \in \Omega} \left| \sum_{x_j \in \mathcal{X}} y_j M^i_{\lambda_s}(x) \right| \leq \max_{x \in \Omega} \sum_{x_j \in \mathcal{X}} |y_j M^i_{\lambda_s}(x)| \]

\[ \leq \max_{x \in \Omega} \sum_{x_j \in \mathcal{X}} |y_j| \cdot |M^i_{\lambda_s}(x)| \leq P^s_\infty ||Y||_\infty \quad \text{where} \quad P^s_\infty = \max_{x \in \Omega} \sum_{j=1}^n |M^j_{\lambda_s}(x)| \]

Now, for establishing bounds on \( P^s_\infty \), we proceed as follows. Let \( x^* \in \Omega \) be the data point at which the \( \sum_{j=1}^n |M^j_{\lambda_s}(x^*)| \) is maximized.

\[ P^s_\infty = \sum_{j=1}^n |M^j_{\lambda_s}(x^*)| = \sum_{j=1}^n |\delta^x_{x^*} M^j_{\lambda_s}| \leq \sum_{j=1}^n ||\delta^x_{x^*}||_{\mathcal{H}_s^*} ||M^j_{\lambda_s}||_{\mathcal{H}_s} = \sum_{j=1}^n ||M^j_{\lambda_s}||_{\mathcal{H}_s} \]

The last equality here comes from the assumed normalization: \( ||\delta^x||_{\mathcal{H}_s^*}^2 = 1 \).

Using the expression for \( M^j_{\lambda_s} \) from Theorem 2

\[ (3.23) \quad < M^j_{\lambda_s}, M^j_{\lambda_s} >_{\mathcal{H}_s} = c^T_j (B^s)^T (B^s + nP^Q_s)^{-1} R_s |x_j R^T_s |x_j (B^s)^T (B^s + nP^Q_s)^{-1} B^s e_j \]
\begin{align*}
(3.24) & \quad = e_j^T B^s (B^s B^s + nP_s^Q)^{-1} B^s (B^s B^s)^{-1} R_s |_{X_s}, \\
& \quad \quad \quad \quad R_s |_{X_s} B^s (B^s B^s)^{-1} (B^s B^s)^{-1} R_s |_{X_s} = M_0
\end{align*}

Now realizing
\[ B^s (B^s B^s + nP_s^Q)^{-1} B^s = U(\Lambda_s) \] and \[ B^s (B^s B^s)^{-1} R_s |_{X_s} = M_0 \]

we get
\[ < M^j_{\Lambda_s}, M^i_{\Lambda_s} >_{\mathcal{H}_s} = e_j^T U(\Lambda_s) M_0 M_0^T U^T(\Lambda_s) e_j \]

If, \( U_j(\Lambda_s) \) represents the \( j^{th} \) influence vector, then we get
\[ < M^j_{\Lambda_s}, M^i_{\Lambda_s} >_{\mathcal{H}_s} = | < U_j(\Lambda_s), M_0 > |^2 \implies ||M^j_{\Lambda_s}||_{\mathcal{H}_s} = | < U_j(\Lambda_s), M_0 > | \]

Therefore we get the upper bound on \( P^s_{\infty} \) as
\begin{equation}
(3.25) \quad P^s_{\infty} \leq \sum_{j=1}^{n} | < U_j(\Lambda_s), M_0 > | = ||U(\Lambda_s) M_0||_1
\end{equation}

For computing the lower bound, we again begin with the fact that,
\[ P^s_{\infty} = \max_{x \in \Omega_s} ||M_{\Lambda_s}(x)||_1 \geq \max_{x \in \Omega_s} ||M_{\Lambda_s}(x)||_2 = ||M_{\Lambda_s}||_2 \]

However from the computations for upper bound and Theorem 2 we infer
\[ < M_{\Lambda_s}, M_{\Lambda_s} >_{\mathcal{H}_s} = ||U(\Lambda_s) M_0||_2^2 \]

Thus establishing the stated theorem \( \square \)

Proceeding further we provide a result which bounds the approximation produced by the proposed approach at any data point \( x \in \Omega_s \) and scale \( s \)

**Corollary 5.1.** The approximation at any \( x \in \Omega_s \) is bounded in the sense
\begin{equation}
(3.26) \quad |A_s f(x)| \leq ||U(\Lambda_s) M_0||_1 ||Y||_\infty
\end{equation}

**Proof.** The proof follows similar steps to the previous theorem. Beginning with the inner product representation of the approximation
\[ |A_s f(x)| = \sum_{j=1}^{n} |y_j| M_{\Lambda_s}^j(x) \leq \sum_{j=1}^{n} |y_j| \cdot ||M_{\Lambda_s}^j(x)||_1 \leq ||Y||_\infty \sum_{j=1}^{n} ||M_{\Lambda_s}^j(x)||_1 \]

Thus, by referring to the upper bound in Theorem 5 the result follows \( \square \)

Now, as stated earlier, we provide bounds for the error in approximation at any new data point

**Theorem 6.** The pointwise approximation error (in definition 1) for any \( x \in \Omega_s \)
\begin{equation}
(3.27) \quad Error(x) = |f(x) - E[A_s f](x)| = |\delta_s^*(f) - M_{\Lambda_s}^T(x) \delta_s^*(f)| = |E_s^*(f)|
\end{equation}

follows the upper bound
\[ Error(x) \leq (1 - a) ||f||_{\mathcal{H}_s} \]

Where \( a = M_0^T(x) U(\Lambda_s) R_s(x) \)
Proof. Starting with the the optimal value of $M_\Lambda(x)$ obtained in Theorem \ref{thm:main}

\begin{equation}
M_\Lambda(x) = B^s(B^{sT}B^s + nP_s^Q)^{-1}R(x)|_x
\end{equation}

Substituting it in the squared error functional norm

$$
\|E_{\Lambda_x}^x\|^2_{|H_x^s} = \langle \delta_x^s - M_\Lambda^T(x)\delta_x^s, \delta_x^s - M_\Lambda^T(x)\delta_x^s \rangle > 0
$$

Starting with the second term

$$
M_\Lambda^T(x)R_s(x) = R(x)|_{X_s}^T(B^{sT}B^s + nP_s^Q)^{-1}B^{sT}R_s(x)
$$

Coming to the third term, $M_\Lambda^T(x)G_sM_\Lambda^s(x)$

$$
= R(x)|_{X_s}^T(B^{sT}B^s + nP_s^Q)^{-1}B^{sT}G_sB^{sT}B^s + nP_s^Q)^{-1}B^{sT}R_s(x)
$$

Therefore

\begin{equation}
\|E_{\Lambda_x}^x\|^2_{|H_x^s} = 1 - 2a + a^2 \quad \text{where } a = M_\Lambda^T(x)U(\Lambda_s)R_s(x)
\end{equation}

Now, coming back to the single point evaluation error representation as discussed earlier

\begin{equation}
Error(x) = |E_{\Lambda_x}^x(f)| \leq \|E_{\Lambda_x}^x\|_{|H_x^s}||f||_{|H_x^s}
\end{equation}

Hence the result follows.

\[ \square \]

4. Results

In this section, we present the results of the proposed hierarchical approach on univariate and multivariate synthetic datasets \cite{5} along with performance analysis on a time series dataset from remote sensing literature \cite{3}. This makes sense as simulated datasets can test the modeling capability with respect to the truth and application on real datasets can test the behavior of the proposed method on the challenges which come with the real observations.

Firstly we begin with the application on two test functions (shown in Figure \ref{fig:results}). The univariate function here shows noisy data sampled from the 1-d Schwefel function \cite{15} in (a) and (b) . The non-convexity of this function coupled with sharp curvature changes is expected to pose a good challenge for any noisy data modeling procedure. The multivariate function here ((c) and (d)) pose similar challenges but in higher dimensions.
Figure 2. (a) and (b): Univariate test function. (a) shows the 1-d Schwefel function along with the sampled noisy data. (b) here just shows this sampled data to give a visual intuition. (c) and (d): Multivariate test function. Here we show the bi-variate Bohachevsky function which we use for our analysis. Plot in (c) shows the true function whereas the plot (d) shows the noisy data sampled from it.

4.1. Understanding the behavior with scales. Considering the univariate test function, Figure 3 provides an intuitive understanding of the behavior of the approach across the scales. Here, starting with scale 0, we show that at each scale increment, more and more points are chosen in the sparse representation leading to the corresponding improvement in the produced approximation. Here we also compute compression ratio at scale $s$ ($comp_s$) defined as

$$comp_s = 1 - \frac{l_s}{n}$$

where $l_s$ is the cardinality of $X_s$.

Therefore a value of $comp_s$ closer to 1 shows that very few observations were selected in the sparse representation and hence represents good compression being achieved. Starting with scale 0 (Figure 3), the cost of fitting quantified as the optimal GCV value was observed to achieve a minima at scale 7 (details are shown in Table 1), establishing it as the convergence scale (t). This is also evident from the quality of the approximation produced at scale 7 (in Figure 3). Moreover it should be noted that the cost of fitting at convergence scale was even less than cost of fitting with the full datasets (Table 1). This is intuitive since here we are trying to find a trade-off between model complexity and generalization capability.
Table 1. Performance of the proposed approach on Univariate (1d Schwefel) and Multivariate (Bohachevsky) test function. For Univariate test function, we have shown the compression ratio $comp_s$ (4.1), optimal cost at scale $s$ (2.19) and optimal penalty order $q$ for all scales (0 to 11 as shown in Figure 3). For the Multivariate test function, the same analysis has been shown (with scales going from 0 to 9 as shown in Figure 4). The optimal penalties in X and Y direction is denoted by $q_{opt}^x$ and $q_{opt}^y$ respectively. Overall the scale with the minimum fitting cost ($Cost_s$) is highlighted (t = 7) for both cases.

4.2. Application on real data. Here we consider the application of our approach on time series of cm. equivalents of water height. These time series were derived in [33] with the objective of studying changes in mass of ice around the globe (with regions divided broadly as ice sheets, ice shelves, land and water). For our purpose we consider 4 different time series here as shown in Figure 5. Here time series 1 and 2 are from Greenland showing the accumulation and ablation (melting) behavior respectively. Time series 3 and 4 show this behavior for Antarctic ice sheet. Figure 6 then shows the approximation produced by our approach on these time series.
For time series 1, the approach is able to capture a rich structure from previous noisy looking data. Here one other important thing to note is that all the points were selected in the sparse representation to produce the best possible approximation. This further shows the nature of the approach to prefer good approximation over a simpler model. For time series 2, we have a clear periodicity in the structure of the data which is suitably captured by our approach. Moving further, time series 3 again shows one very important property of our approach. Here since the data is very noisy, hence the sparse representation chosen is very small as compared to the full dataset. This is because of the lack of structure in the data and hence a simpler model leads to a better generalization performance. In the last time series (time series 4), the algorithm again captures the periodicity in the data while choosing a subset of the dataset as the optimal sparse representation for generating approximations. The compression ratios and the optimal penalty order for the test time series are shown in Table 2.
Figure 4. Scale-wise performance and solution of proposed approach on bi-variate test function. The light surface above and below the mean approximation (magenta colored) shows the 95% t-confidence intervals.
Figure 5. Location of the test time series on the Greenland (time series 1 in (a) and 2 in (b)) and Antarctic (time series 3 in (c) and 4 in (d)) ice sheets. We have chosen the time series from both accumulation (near the center with more frequent snowing) and ablation zones (near the edge with higher degree of fluctuations and activity) of the ice sheets for testing the proposed approach.

Figure 6. Performance on the 4 time series from Greenland and Antarctic Icesheets (shown in Figure 5) with 95% t-confidence intervals.

5. Conclusion

In this paper, we presented a hierarchical regularization network based approach to generate sparse representations for noisy datasets with Generalized Cross Validation (GCV) for model selection and fitting. We provided a detailed theoretical
framework for the approach particularly studying the approximation behavior coupled with consistency and convergence.

These sparse representations were also shown to act as a model for the datasets to produce good approximations at previously un-observed data points. For testing the procedure, test datasets were picked from both simulations and observed real data repositories. On all of these datasets the approach was found to perform well providing an inference for the approximation with confidence intervals from the generated sparse representations.

The next steps of this approach to sparse modeling with data reduction will be to extend the approach to very large datasets through efficient distributed implementations and intelligent data structures. The quantification of model uncertainty could also be further improved by Bayesian sampling approaches that can effectively propagate the uncertainty of scale selection and inference of other parameters to the final model outcome. These are expected to be a part of our future works.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
Feature & Time Series & TS1 & TS2 & TS3 & TS4 \\
\hline
Convergence Scale \((t)\) from 0 to 10 & & 10 & 9 & 1 & 9 \\
Compression ratio at \(s=t\) & & 0.00 & 0.22 & 0.91 & 0.22 \\
Optimal Penalty at \(s=t\) \((q^{opt})\) & & 2 & 2 & 1 & 1 \\
\hline
\end{tabular}
\caption{Performance details on the time series data (Figure 5).}
\end{table}

Here we have used the acronym TS for Time Series.

\section*{References}
1. William K Allard, Guangliang Chen, and Mauro Maggioni, \textit{Multi-scale geometric methods for data sets ii: Geometric multi-resolution analysis}, Applied and Computational Harmonic Analysis \textbf{32} (2012), no. 3, 435–462.
2. Nachman Aronszajn, \textit{Theory of reproducing kernels}, Transactions of the American mathematical society \textbf{68} (1950), no. 3, 337–404.
3. L Mark Berliner, \textit{Hierarchical bayesian time series models}, Maximum entropy and Bayesian methods, Springer, 1996, pp. 15–22.
4. Amit Bermanis, Amir Averbuch, and Ronald R Coifman, \textit{Multiscale data sampling and function extension}, Applied and Computational Harmonic Analysis \textbf{34} (2013), no. 1, 15–29.
5. Bastian Bohn, Jochen Garcke, and Michael Griebel, \textit{A sparse grid based method for generative dimensionality reduction of high-dimensional data}, Journal of Computational Physics \textbf{309} (2016), 1–17.
6. Nunzio Alberto Borghese and Stefano Ferrari, \textit{Hierarchical rbf networks and local parameters estimate}, Neurocomputing \textbf{19} (1998), no. 1-3, 259–283.
7. William L Briggs, Steve F McCormick, et al., \textit{A multigrid tutorial}, vol. 72, Siam, 2000.
8. Martin D Buhmann, \textit{Radial basis functions: theory and implementations}, vol. 12, Cambridge university press, 2003.
9. D Chaudhuri, CA Murthy, and BB Chaudhuri, \textit{Finding a subset of representative points in a data set}, IEEE transactions on systems, man, and cybernetics \textbf{24} (1994), no. 9, 1416–1424.
10. Guangliang Chen, Anna V Little, and Mauro Maggioni, \textit{Multi-resolution geometric analysis for data in high dimensions}, Excursions in Harmonic Analysis, Volume 1, Springer, 2013, pp. 259–285.
11. Ronald R Coifman, Stephane Lafon, Ann B Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven W Zucker, \textit{Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods}, Proceedings of the National Academy of Sciences \textbf{102} (2005), no. 21, 7432–7437.
12. Ronald R Coifman and Mauro Maggioni, *Diffusion wavelets*, Applied and Computational Harmonic Analysis **21** (2006), no. 1, 53–94.
13. Noel Cressie and Christopher K Wikle, *Statistics for spatio-temporal data*, John Wiley & Sons, 2015.
14. Ireneusz Czarnowski and Piotr Jedrzejowicz, *An approach to data reduction for learning from big datasets: Integrating stacking, rotation, and agent population learning techniques*, Complexity **2018** (2018).
15. Ingrid Daubechies, *Ten lectures on wavelets*, vol. 61, Siam, 1992.
16. Carl De Boor, Carl De Boor, Etats-Unis Mathématicien, Carl De Boor, and Carl De Boor, *A practical guide to splines*, vol. 27, springer-verlag New York, 1978.
17. Stefano De Marchi and Robert Schaback, *Stability of kernel-based interpolation*, Advances in Computational Mathematics **32** (2010), no. 2, 155–161.
18. Paul HC Eilers and Brian D Marx, *Flexible smoothing with b-splines and penalties*, Statistical science (1996), 89–102.
19. Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio, *Regularization networks and support vector machines*, Advances in computational mathematics **13** (2000), no. 1, 1.
20. Gregory E Fasshauer and Jack G Zhang, *Preconditioning of radial basis function interpolation systems via accelerated iterated approximate moving least squares approximation*, Progress on Meshless Methods, Springer, 2009, pp. 57–75.
21. Stefano Ferrari, Mauro Maggioni, and N Alberto Borghese, *Multiscale approximation with hierarchical radial basis functions networks*, IEEE Transactions on Neural Networks **15** (2004), no. 1, 178–188.
22. Frédéric Ferraty and Philippe Vieu, *Nonparametric functional data analysis: theory and practice*, Springer Science & Business Media, 2006.
23. Michael S Floater and Armin Iske, *Multistep scattered data interpolation using compactly supported radial basis functions*, Journal of Computational and Applied Mathematics **73** (1996), no. 1-2, 65–78.
24. Malcolm R Forster, *Key concepts in model selection: Performance and generalizability*, Journal of mathematical psychology **44** (2000), no. 1, 205–231.
25. Meirav Galun, Ronen Basri, and Irad Yavneh, *Review of methods inspired by algebraic-multigrid for data and image analysis applications*, Numerical Mathematics: Theory, Methods and Applications **8** (2015), no. 2, 283–312.
26. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.
27. Peter J Green and Bernard W Silverman, *Nonparametric regression and generalized linear models: a roughness penalty approach*, CRC Press, 1993.
28. Michael Griebel and Alexander Hullmann, *A sparse grid based generative topographic mapping for the dimensionality reduction of high-dimensional data*, Modeling, Simulation and Optimization of Complex Processes-HPSC 2012, Springer, 2014, pp. 51–62.
29. Philipp Griebsch, Dmytro Perekrestenko, Dennis Elbrchter, and Helmut Blcskei, *Deep neural network approximation theory*, 2019.
30. Tailen Hsing and Randall Eubank, *Theoretical foundations of functional data analysis, with an introduction to linear operators*, John Wiley & Sons, 2015.
31. Armin Iske, *Scattered data approximation by positive definite kernel functions*, Rend. Sem. Mat. Univ. Pol. Torino **69** (2011), no. 3, 217–246.
32. Dan Kushnir, Meirav Galun, and Achi Brandt, *Efficient multilevel eigensolvers with applications to data analysis tasks*, IEEE transactions on pattern analysis and machine intelligence **32** (2009), no. 8, 1377–1391.
33. Scott B Luthcke, TJ Sabaka, BD Loomis, AA Arendt, JJ McCarthy, and J Camp, *Antarctica, greenland and gulf of alaska land-ice evolution from an iterated grace global mascon solution*, (2013).
34. Mauro Maggioni, James C Bremer Jr, Ronald R Coifman, and Arthur D Szlam, *Biorthogonal diffusion wavelets for multiscale representation on manifolds and graphs*, Wavelets XI, vol. 5914, International Society for Optics and Photonics, 2005, p. 59141M.
35. Stephane G Mallat, *A theory for multiresolution signal decomposition: the wavelet representation*, IEEE Transactions on Pattern Analysis & Machine Intelligence (1989), no. 7, 674–693.
36. J Tinsley Oden and Leszek Demkowicz, *Applied functional analysis*, Chapman and Hall/CRC, 2017.
37. Nathan D Pearce and Matthew P Wand, *Penalized splines and reproducing kernel methods*, The american statistician **60** (2006), no. 3, 233–240.
38. Tomaso Poggio and Federico Girosi, *Networks for approximation and learning*, Proceedings of the IEEE **78** (1990), no. 9, 1481–1497.
39. ______, *Regularization algorithms for learning that are equivalent to multilayer networks*, Science **247** (1990), no. 4945, 978–982.
40. Carl Edward Rasmussen, *Gaussian processes in machine learning*, Advanced lectures on machine learning, Springer, 2004, pp. 63–71.
41. David Ruppert, Matt P Wand, and Raymond J Carroll, *Semiparametric regression*, vol. 12, Cambridge university press, 2003.
42. Yousef Saad, *Iterative methods for sparse linear systems*, vol. 82, siam, 2003.
43. Prashant Shekhar and Abani Patra, *Hierarchical data reduction and learning*, 2019.
44. Klaus Stüben, *A review of algebraic multigrid*, Numerical Analysis: Historical Developments in the 20th Century, Elsevier, 2001, pp. 331–359.
45. S. Surjanovic and D. Bingham, *Virtual library of simulation experiments: Test functions and datasets*, Retrieved November 14, 2019, from [http://www.sfu.ca/~ssurjano](http://www.sfu.ca/~ssurjano).
46. Javier Tejada, Mikhail Alexandrov, Gabriella Skitalinskaya, and Dmitry Stefanovsky, *Selection of statistically representative subset from a large data set*, Iberoamerican Congress on Pattern Recognition, Springer, 2016, pp. 476–483.
47. Robert Tibshirani, Martin Wainwright, and Trevor Hastie, *Statistical learning with sparsity: the lasso and generalizations*, Chapman and Hall/CRC, 2015.
48. Muhammad Habib ur Rehman, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah, and Samee U Khan, *Big data reduction methods: a survey*, Data Science and Engineering **1** (2016), no. 4, 265–284.
49. Vladimir Vapnik, *The nature of statistical learning theory*, Springer science & business media, 2013.
50. Grace Wahba, *Spline models for observational data*, vol. 59, Siam, 1990.
51. Holger Wendland, *Scattered data approximation*, vol. 17, Cambridge university press, 2004.
52. Ahmet Artu Yıldırım, Cem Özdoğan, and Dan Watson, *Parallel data reduction techniques for big datasets*, Big Data: Concepts, Methodologies, Tools, and Applications, IGI Global, 2016, pp. 734–756.

**DATA INTENSIVE STUDIES CENTER, TUFTS UNIVERSITY, MEDFORD, MA, 02155**
**E-mail address:** prashant.shekhar@tufts.edu

**DATA INTENSIVE STUDIES CENTER, DEPARTMENT OF MATHEMATICS, DEPARTMENT OF COMPUTER SCIENCE, TUFTS UNIVERSITY, MEDFORD, MA, 02155**
**E-mail address:** abani.patra@tufts.edu