Available online at www.sciencedirect.com

ScienceDirect

Journal of Approximation Theory 245 (2019) 101–129

JOURNAL OF Approximation Theory

www.elsevier.com/locate/jat

Full Length Article

Approximation of generalized ridge functions in high dimensions

Sandra Keiper

Department of Mathematics, Technische Universität Berlin, 10623 Berlin, Germany

Received 24 January 2017; received in revised form 6 February 2019; accepted 12 April 2019
Available online 22 May 2019

Communicated by D.-X. Zhou

Abstract

This paper studies the approximation of generalized ridge functions, namely of functions which are constant along some submanifolds of $\mathbb{R}^N$. We introduce the notion of linear-sleeve functions, whose function values only depend on the distance to some unknown linear subspace L. We propose two effective algorithms to approximate linear-sleeve functions $f(x) = g(\text{dist}(x, L)^2)$, when both the linear subspace $L \subset \mathbb{R}^N$ and the function $g \in C^s[0, 1]$ are unknown. We will prove error bounds for both algorithms and provide an extensive numerical comparison of both. We further propose an approach of how to apply these algorithms to capture general sleeve functions, which are constant along some lower dimensional submanifolds.

Keywords: Ridge functions; Function approximation; Big data; High dimensions; Active variables; Active subspaces; Optimization over Grassmannian manifolds

1. Introduction

Nowadays we are living in a world where the acquisition, analysis and storage of big data play a major role. Usually data is modeled as functions $f : X \to Y$, where X can be $\mathbb{R}^N$ or a general curved surface. In particular the approximation of such functions from point queries, when N is very large, is an important field. Such problems arise, for example, in learning theory [22], in modeling physical and biological systems [17], as well as neural networks [4] and in parametric and stochastic PDEs [7].

E-mail address: keiper@math.tu-berlin.de.

https://doi.org/10.1016/j.jat.2019.04.006

Because of the so-called curse of dimensionality, a notion introduced in 1961 by Richard Bellman [3], the handling of functions in many variables is an ambitious task. Namely, functions on $\mathbb{R}^N$ with smoothness of order s can in general be recovered with an accuracy of at most $n^{-s/N}$, using at most n function evaluations. Thus, the learning of functions depending on a large number of variables is particularly difficult even with smoothness assumptions on f [9,10,19]. Certainly, we need to impose additional structure on f to achieve efficient learning [6,14,16,20,21].

1.1. Ridge functions

One popular approach to break the curse of dimensionality is to consider ridge functions of the form

$$\mathbb{R}^N \supseteq \Omega \ni x \mapsto f(x) = g(Ax), \tag{1}$$

where $A \in \mathbb{R}^{m\times N}$, with m considerably smaller than N, is usually called *ridge matrix* and $g \in C^s(\mathbb{R}^m)$, $1 \le s \le 2$, is called the *ridge profile*. The requirement for the function to have at least one derivative is essential. In fact, it was shown in [18] that ridge functions need to have a first derivative uniformly bounded away from zero in the origin in order to reduce the complexity of the approximation task.

For particular choices of A different approaches to successfully learn ridge functions have been investigated. For example, if A is of the form $A^T = [e_{i_1}, \dots, e_{i_m}]$, for $e_{i_k} \in \mathbb{R}^N$ being the canonical unit vectors and $i_k \in \{1, \dots, N\}$, f can be rewritten as a function which depends only on a few variables, i.e., $f(x_1, \dots, x_N) = g(x_{i_1}, \dots, x_{i_m})$. An approach to recover the active variables and approximating the ridge profile g has been given in [11]. It was shown that by adaptive sampling we can obtain similar estimates as if the active coordinates $i_1, \dots, i_m$ are known to us.

Another special case of (1) is to assume that $m = 1$ and that the matrix A is therefore a vector, usually called *ridge vector* and denoted by $A^T =: a$. In this case, f is of the form

$$f(x) = g(\langle x, a\rangle). \tag{2}$$

The recovery of such ridge functions from point queries was first considered by Cohen, Daubechies, DeVore, Kerkyacharian, and Picard in [6] for ridge functions with a positive ridge vector. It was shown that the accuracy of their method is close to the approximation rate of one-dimensional continuous functions.

However, the algorithm from [6] does not apply to arbitrary ridge vectors. In [14,16,21] new algorithms were introduced to waive the assumption of a positive ridge vector. The main idea of the algorithm in [16] is to approximate the gradient of f by divided differences, exploiting the fact that the gradient of f is some scalar multiple of the ridge vector. The accuracy of the approximation of the gradient is determined by the choice of the step size in the computation of the divided differences, whereas the number of sampling points is fixed.

The approach by Fornasier, Schnass and Vybiral [14] is rather based on compressed sensing and applies to (1) very generally. Thus, not the gradient but the directional derivatives of f were approximated at a certain number of random points in random directions. However, especially for the methods in [6,14], the authors need a restrictive assumption to use compressed sensing techniques. That is, the ridge vector a can be well-approximated by a sparse subset of its coefficients. In [21] this assumption could be removed by leveraging the Dantzig selector [5] to recover an approximation of a.

However, the structure assumption on f to be a ridge function can be very restrictive. If we consider, for example a sensor network, where we have a certain number of sensors, say N, which measure the moisture, temperature and pressure to forecast forest fire, the aim is to compute the risk of fire by a function $f : \mathbb{R}^{3N} \to \mathbb{R}$ depending on the measurements of the sensors. It is then very unlikely that the combination of measurements which yield the same risk of fire lie on a $3N-1$-dimensional hyperplane, since also parameters like topography and vegetation influence the prediction. Much more likely is the assumption that these combinations lie on a lower dimensional manifold.

1.2. Sleeve functions

To allow for the recovery of more general functions, which are constant along some lower-dimensional submanifolds, we will introduce the notion of *sleeve functions* and as a special case of *linear-sleeve* functions. Within this paper we will then investigate and analyze algorithms to capture linear-sleeve functions and we will propose a technique to apply these methods to general sleeve functions.

Definition 1.1. Let $g \in C^s[-r, r]$, $r \in \mathbb{R}_+$, $M \subset \mathbb{R}^N$ a d-dimensional smooth submanifold of $\mathbb{R}^N$, and $\operatorname{tub}_r(M) := \{x \in \mathbb{R}^N : \operatorname{dist}(x, M) < r\}$, then we call $f : \operatorname{tub}_r(M) \to \mathbb{R}$ a *sleeve function* if we can rewrite f in terms of g by

$$f(x) = g(\operatorname{dist}(x, M)^2), \tag{3}$$

for $x \in \operatorname{tub}_r(M)$. In the case where M is a linear subspace, we call f a *linear-sleeve function* and denote $L := M$, to emphasize the special case, i.e., we write:

$$f(x) = g(\operatorname{dist}(x, L)^2). \tag{4}$$

The need to restrict g to a bounded domain is twofold; on the one hand, if we wish to recover g from a finite number of sampling points, we do need this restriction and on the other hand, it is useful for the approximation task to have a unique mapping $x \mapsto x_0$ with $\operatorname{dist}(x, M) = \operatorname{dist}(x, x_0)$. Thus, in the case of M being a linear subspace, r can be chosen arbitrarily, where in the general case r is chosen to be the radius of a non-self-intersecting tube around M. For an illustration of linear-sleeve functions we refer to Fig. 2.

Note that the notion of sleeve functions is indeed a generalization of ridge function, thus, if L is an $N-1$-dimensional subspace, we can rewrite $\operatorname{dist}(x, L)^2 = \langle x, a\rangle^2$, where a is the normal vector of L. Also note that this formulation is very different from the one introduced in [14]. Indeed, if f is of the form $f = g(A\cdot)$, the level sets are linear subspaces, whereas this is not true for linear-sleeve functions (cf. Fig. 2).

Furthermore, observe that even by separating the approximation task in approximating g and M, we cannot simply use manifold learning algorithms to approximate M, since manifold learning algorithms (cf. e.g. [2,8]) usually assume that we can sample from the manifold. However, we need to reconstruct the level sets (or at least one, namely M) without knowing in advance to which level set the sampling points belong; actually it is very likely that all sampling points belong to different level sets.

1.3. Our contribution

Our work studies the approximation of linear-sleeve function of the form $f(x) = g(\operatorname{dist}(x, L))$ for $x \in \operatorname{tub}_1(L)$, where $g \in C^s[0, 1]$ is monotone and $L \subset \mathbb{R}^N$ is a d-dimensional

subspace of $\mathbb{R}^N$ under additional assumptions on the profile g (see Fig. 1). In particular we will assume that the derivative of g is bounded. We will provide and analyze two different algorithms to capture linear sleeve functions from point queries. Our main contributions can be summarized as follows.

- *Adaptive Algorithm.* The first algorithm, to which we refer to as *ATPE*, is based on the fact that the gradient of f in some $x \in \mathbb{R}^N$ is orthogonal to the level set of f in x. We will show that the restriction of f to the plane, which is orthogonal to the gradient and which is the tangent plane of the corresponding level set, is again a linear-sleeve function. We will then argue that applying the same fact iteratively to the restrictions of f, the tangent plane computed in the $N-d$th step gives a reconstruction of L. In ATPE we will then substitute the gradient by divided differences, because we cannot compute the gradient by point queries of f.
- *Optimization Algorithm.* The second algorithm, to which we refer to as *OGM*, is based on a minimization over the Grassmannian manifold. Namely, it will define an objective function, whose minimizer is L. However, we will see that we cannot define this objective functions using only point samples of f. In OGM we therefore approximate this objective function by an objective function whose minimizer $\tilde{L}$ will be proven to be close to L.
- *Error Bounds.* Those two algorithms are of a rather different nature. Whereas the approximation success of the first algorithm depends only on the error of the gradient approximation by divided differences, the success of the second algorithms depends on the error of the approximation of the objective function. The first main theorem states that the error of the approximation of the d-dimensional subspace L using ATPE which is a randomized algorithm can with high probability be bounded by

  $$\|L - \tilde{L}\|_{\mathrm{HS}} \leq C(1+K)^{N-d}\sqrt{N-d}h^s,$$

  where $\tilde{L}$ is the approximation of L, h can be chosen arbitrarily small but fixed, C, K are some positive constants which depend on the considered function class and the number of function evaluations is given by $(N+1)(N-d)$. For the second main theorem we prove that using OGM which also is a randomized algorithm the approximation error almost surely is given by

  $$\|L - \tilde{L}\|_{\mathrm{HS}} \leq \tilde{C}\sqrt{N-d}M^{-1},$$

  where M is the number of function samples and $\tilde{C}$ a constant only depending polynomially on the dimension of the space. Note that OGM, differently to ATPE, yields a reconstruction error which decreases with the number of sampling points and is not constrained by a fixed number of sampling points, and is therefore advantageous. However, the first algorithm is more promising to apply also to the manifold case.

We further believe that our results will have some impact on the approximation of general sleeve functions of the type (3). Due to the fact that we would need to optimize over all possible d-dimensional submanifolds, to approximate general sleeve functions in a similar way as proposed by OGM, we anticipate that an adaptation of ATPE is more promising.

We believe that one can also use gradient approximations to capture general sleeve functions of the type (3). Roughly said, we propose to use the gradients to compute samples from the manifold. More precisely, knowing the gradient of f at some point x, again would enable us to approximate the sleeve profile g and, under additional assumption, we could use the direction given by the gradient and the value of f in x to translate x to the manifold. A careful estimation

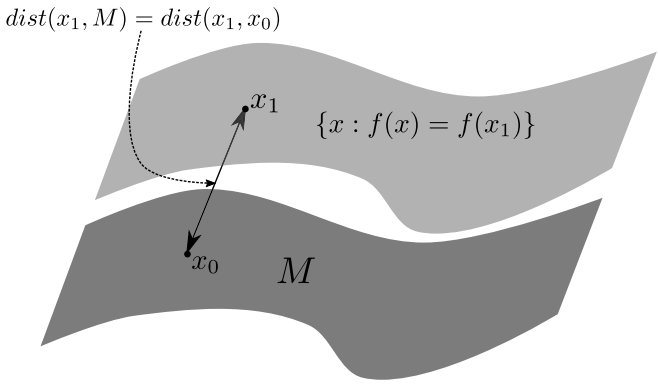


Fig. 1. Generalized ridge function of the form $f(x) := g(\mathrm{dist}(x, M))$.

of the distribution of the translated sample points should then enable us to apply manifold learning algorithms (e.g [2]) to estimate the manifold M.

Note that we will not study the question of stability of the algorithms in this paper. The numerical experiments indicate that the algorithms might be stable in a certain sense. However, an intensive numerical and theoretical analysis remains open for future work.

The paper is organized as follows: After introducing some preliminaries, we will present and analyze ATPE in Section 3. In Section 4 we will introduce and analyze OGM. The consideration will be completed in Section 5 by some promising numerical results.

2. Preliminaries

To put our results in a precise setting, we introduce the class $\mathcal{LR}(s, d)$ of all linear-sleeve functions $f(x) := g(\mathrm{dist}(x, L)^2)$, where $g \in C^s[0, 1]$ and $L \subset \mathbb{R}^N$ a d-dimensional subspace, i.e., $d \in [N] := \{1, \ldots, N\}$. We use the following norm, subsequently referred to as *Hölder norm*, on C^s. For $k < s \leq k + 1$, with $k \in \mathbb{N}$, we define

$$\|g\|_{C^s} := \|g\|_{C^s[0,1]} := |g^{(k)}|_{C^{(s-k)}} + \sum_{j=0}^{k} \|g^{(j)}\|_{C[0,1]},$$

where $g^{(j)}$ denotes the jth derivative of g, and, for $0 < \beta \leq 1$, we set

$$|g|_{C^\beta} := \sup_{x \neq y} \frac{|g(x) - g(y)|}{|x - y|^\beta}.$$

Note that we call g *Lipschitz continuous* if $|g|_{C^1}$ is bounded. We then call $|g|_{C^1}$ *Lipschitz constant* or *Lipschitz norm* of g. The ith coordinate of a vector x will be denoted by x_i. If the vector itself already has an index, e.g. x_j, we denote the ith coordinate of x_j by x_{ji}. If we want to highlight the dimension of the vector space, we sometimes write $\|\cdot\|_{\ell_p^N} := \|\cdot\|_p$ for the ℓ_p norm of a vector, for $p = 1, 2$. The weak-ℓ_p norm of a vector $x \in \mathbb{R}^N$ is the smallest constant M, such that

$$\#\{i : x_i \geq \varepsilon\} \leq M\varepsilon^{-1/p}, \quad \varepsilon > 0.$$

We further recall the following useful property of any norm on $\mathbb{R}^N$.

Lemma 2.1 ([16]). *Let $\|\cdot\|$ be any norm on $\mathbb{R}^N$ and $x \in \mathbb{R}^N$ with $\|x\| = 1$, $\tilde{x} \in \mathbb{R}^N \setminus \{0\}$ and $\lambda \in \mathbb{R}$. Then*

$$\|\operatorname{sign}(\lambda)\frac{\tilde{x}}{\|\tilde{x}\|} - x\| \leq \frac{2\|\tilde{x} - \lambda x\|}{\|\tilde{x}\|}.$$

We will denote the ith canonical unit vector, with a one in the ith coordinate and zero elsewhere, by e_i. The Grassmannian manifold of all d-dimensional subspaces of $\mathbb{R}^N$ is denoted by $G(d, N)$ and for simplicity we will denote the orthogonal projection $P_H x$ of a vector $x \in \mathbb{R}^N$ to a subspace $H = \operatorname{span}\{u_1, \ldots, u_d\} \in G(d, N)$ by Hx. This notation relates to the matrix representation of an orthogonal projection given by $H = \sum_{i=1}^{d} u_i u_i^T$. For a bounded operator $L : \mathbb{R}^N \to \mathbb{R}^N$, the Hilbert–Schmidt norm is given by

$$\|L\|_{\mathrm{HS}} = \sqrt{\sum_{i=1}^{N} \|Le_i\|_2^2}.$$

For two d-dimensional subspaces $L, \tilde{L} \subset \mathbb{R}^N$ the Hilbert–Schmidt distance is given by $\|L - \tilde{L}\|_{\mathrm{HS}} := \|P_L - P_{\tilde{L}}\|_{\mathrm{HS}}$, i.e., by the Hilbert–Schmidt norm of the difference of the corresponding projections. Further note, that the Hilbert–Schmidt norm $\|L - \tilde{L}\|_{\mathrm{HS}}$ coincides with the Frobenius norm of the corresponding matrix representation, i.e., $\|L - \tilde{L}\|_{\mathrm{HS}} = \|L - \tilde{L}\|_{\mathrm{F}}$. The orthogonal complement of a subspace P is denoted by $P^{\perp}$ and the distance of a vector $x \in \mathbb{R}^N$ to a subspace, respective subset, $P \subset \mathbb{R}^N$ is defined by

$$\operatorname{dist}(x, P) := \min_{y \in P} \|x - y\|_2.$$

In the sequel, for two quantities $A, B \in \mathbb{R}$, which may depend on several parameters, we shall write $A \lesssim B$, if there exists a constant $C > 0$ such that $A \leq CB$, uniformly in the parameters. If the converse inequality holds true, we write $A \gtrsim B$ and if both inequalities hold, we shall write $A \asymp B$.

We will often say that we choose a vector in $\mathbb{R}^N$ randomly. In that case, we mean with respect to a probability measure that has a density with respect to the Lebesgue measure.

Finally, we want to recall some approximation properties of functions in $C[0, 1]$. For two given integers $S > 1$ and $M \geq 2$, we consider the space $\mathcal{S}_{h,S}$, $h := 1/M$, of piecewise polynomials of degree $S - 1$ with equally spaced knots at the points ih, $i = 1, \ldots, M - 1$, and having continuous derivatives of order $S - 2$. It is well-known (cf. e.g. [10]) that there is a class of linear operators Q_h which maps $C[0, 1]$ into $\mathcal{S}_{h,S}$. These operators are usually called *quasi-interpolants*. For a function $g \in C[0, 1]$, the application of a quasi-interpolant only depends on the values of g at the points ih, $i = 0, \ldots, M$. Furthermore, we can choose the operator Q_h to fulfill the following property: For all $g \in C^s([0, 1])$, $0 < s \leq S$,

$$\|g - Q_h g\|_{C[0,1]} \leq D|g|_{C^s[0,1]} h^s, \tag{5}$$

with D, a constant depending only on S [10, pp. 354 ff].

3. An adaptive algorithm estimating tangent planes

An obvious approach to approximate sleeve functions of the form (3) is to apply the methods to recover classical ridge functions of the form (2), i.e., to approximate a linear sleeve function by a classical ridge function. Of course, this method can only provide good approximation results if the sleeve function is close to a classical ridge function in a certain sense, cf. [15].

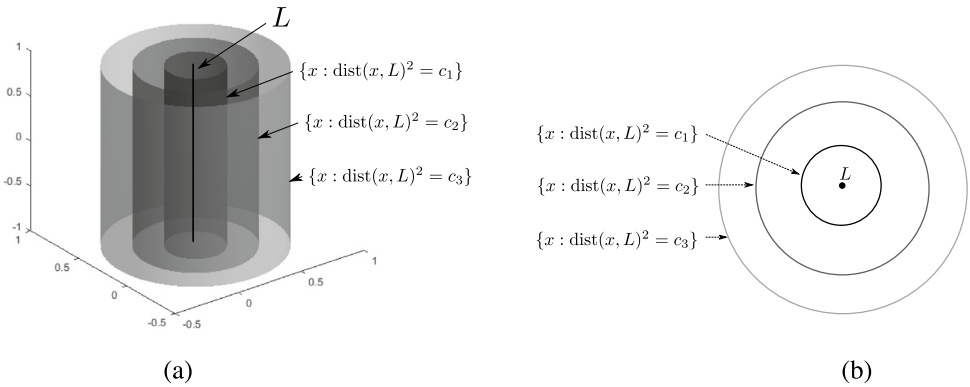


Fig. 2. Generalized ridge function of the form $f(x) := g(\mathrm{dist}(x, L)^2)$, where L is some one-dimensional affine subspace of $\mathbb{R}^N$. The figures show three level sets for the function $f(x_1, x_2, x_3) = x_2^2 + x_3^2 = \mathrm{dist}(x, L)^2$, where $L := \mathrm{span}\,(1, 0, 0)$. (a) The level sets are illustrated in the ambient space $\mathbb{R}^3$. (b) Projection of the level sets onto the subspace orthogonal to L, which is here the xy-plane.

However, it would be more convenient to approximate a function of the form (3) by an estimator of the same form.

We first observe that we can rewrite a linear-sleeve function as

$$f(x) = g(\mathrm{dist}(x, L)^2) = g(\|Px\|_2^2),$$

where P is the orthogonal projection to the $(N-d)$-dimensional subspace P orthogonal to L.

The algorithm, we will introduce in this section, will, similarly as in [6,16], exploit the fact that we can estimate the tangent plane in some $x_0 \in \mathbb{R}^N$ of the $(N-1)$-dimensional submanifold

$$\left\{x \in \mathbb{R}^N : \mathrm{dist}(x, L)^2 = \mathrm{dist}(x_0, L)^2\right\}$$

as the unique hyperplane which is orthogonal to the gradient of f in x_0. We will then show that the function f restricted to this tangent plane is again of the form (4). Of course, we cannot compute the gradient by sampling the function; in the subsequent proposed algorithm we therefore approximate the gradient by divided differences.

3.1. The algorithm

As mentioned before the idea of the first algorithm is to use the fact that the gradient of the linear-sleeve function f in some x_0 is orthogonal to the level sets and that the restriction of f to the corresponding tangent plane is again a linear sleeve function (cf. Fig. 3). We will further show that applying this fact iteratively to restrictions of f finds after $N - d$ steps the wanted subspace L. Hence, the dimension d of the subspace needs to be known. To prove this statement we introduce an adaptive and randomized algorithm to exactly recover the subspace L by computing gradients of f (see Algorithm 1) and show in Theorem 3.1 that this algorithm can recover the subspace L exactly. Thus, our first aim is to show that the system, which is formed by the gradients of the restrictions of f, forms a basis for $P = L^\perp$. Observe that an essential step in this algorithm is to compute gradients of f and is therefore not useful to approximate f from point samples. It only serves as an auxiliary tool to introduce the first main Algorithm 2.

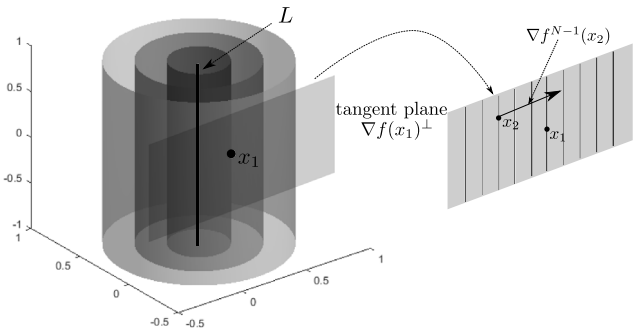


Fig. 3. The restriction of f to the affine subspace which is orthogonal to the gradient of f in some x_1 is again a linear sleeve function. In the illustrated case we can recover L after two steps, because the gradient of the restriction of f in some x_2 is orthogonal to L.

Algorithm 1: ATPC - Approximation by Adaptive Tangent Plane Computation

Data: $f(\cdot) = g(\mathrm{dist}(\cdot, L)^2)$, $\dim(L) = d \in [N]$.
Result: T^d.
begin
  $f^N \longleftarrow f$.
  $T^N \longleftarrow \mathbb{R}^N$.
  for $i = N, \dots, d+1$ **do**
    1. For some randomly chosen, normalized $x_i \in T^i$ compute $\nabla f^i(x_i)$.
    2. $u_i \longleftarrow \nabla f^i(x_i)/\|\nabla f^i(x_i)\|_2$.
    3. $T^{i-1} \longleftarrow (\mathrm{span}\{u_i, \dots, u_N\})^\perp$.
    4. Let f^{i-1} be the restriction of f to T^{i-1}.

As mentioned before this algorithm iteratively computes restrictions of f, such that after $N-d$ steps the restriction of f will be exactly defined on L. Assume we have computed the tangent plane T^i and the restriction f^i of f to the subspace $T^i \subset \mathbb{R}^N$, for $i = N, \dots, d+1$. The algorithm then chooses a point $x_i \in T^i$ randomly and computes the gradient of f^i in x_i (1.). It then normalizes this gradient (2.) and determines the subspace T^{i-1} which is orthogonal to this gradient in T^i (3.). Finally we restrict f^i to T^{i-1} and repeat this procedure until $i = d+1$. The following theorem states that ATPC indeed succeeds to recover L.

Theorem 3.1. *Let $f(x) = g(\mathrm{dist}(x, L)^2) = g(\|Px\|_2^2)$, $g \in C^s[0,1]$, $1 < s \leq 2$, for $x \in tub_1(L)$ and the linear subspace $P^\perp = L \subset \mathbb{R}^N$. Compute T^d as proposed in Algorithm 1. Then L coincides with T^d almost surely.*

Proof. We write L as $L = \mathrm{span}\{u_1, \dots, u_d\}$ where $\{u_1, \dots, u_d, u_{d+1}, \dots, u_N\}$ is an orthonormal basis of $\mathbb{R}^N$, and let $V = [u_1 \dots u_d\, u_{d+1} \dots u_N]$ be the corresponding matrix. We begin by computing the gradient of f and obtain

$$\nabla f(x_N) = 2g'(\|Px_N\|_2^2)Px_N,$$

which is obviously orthogonal to L if $Px_N \neq 0$, which is almost surely true.

Due to the orthogonality of $\nabla f(x_N)$ to L, we can assume that $u_N = \nabla f(x_N)/\|\nabla f(x_N)\|_2$ and then by definition $T^{N-1} = (\text{span}\{u_N\})^\perp = \text{span}\{u_1, \dots, u_{N-1}\}$. We now define f^{N-1} to be the restriction of f to T^{N-1} and $h^{N-1} : \mathbb{R}^{N-1} \to \mathbb{R}$ by

$$h^{N-1}(\hat{x}) := f(V \begin{bmatrix} \hat{x} \\ 0 \end{bmatrix}) = f^{N-1}(V \begin{bmatrix} \hat{x} \\ 0 \end{bmatrix}),$$

for $\hat{x} \in \mathbb{R}^{N-1}$. Then $\nabla h^{N-1^T}(\hat{x}) = \nabla f(x)^T \hat{V}_{N-1}$, where $\hat{V}_i := [u_1 \dots u_i]$, $i = 1, \dots, N$, and $x =: V(\hat{x}, 0)^T \in T^{N-1}$. Thus, the gradient of h^{N-1}, considered as a vector in $\mathbb{R}^N$ is given by

$$\begin{bmatrix} \nabla h^{N-1}(\hat{x}) \\ 0 \end{bmatrix} = V_{N-1}^T \nabla f(x),$$

where $V_i := [u_1 \dots u_i 0 \dots 0]$. We conclude that for some $x_{N-1} \in T^{N-1}$ randomly chosen, we have

$$\nabla f^{N-1}(x_{N-1}) = V V_{N-1}^T \nabla f(x_{N-1}) = 2g'(\|Px_{N-1}\|_2^2) V V_{N-1}^T Px. \tag{6}$$

A straightforward computation shows that $V V_{N-1}^T Px$ is the projection of Px to T^{N-1}. Thus, it is obvious that for some $x_{N-1} \in T^{N-1}$, chosen as in Algorithm 1, $\nabla f^{N-1}(x_{N-1})$ is orthogonal to L, i.e., if $Px_{N-1} \neq 0$, which is almost surely the case.

Further, $\nabla f^{N-1}(x_{N-1})$ is also orthogonal to u_N, since it lies in T^{N-1}. Therefore, we set

$$u_{N-1} = \nabla f^{N-1}(x_{N-1})/\|\nabla f^{N-1}(x_{N-1})\|_2$$

for some $x_{N-1} \in T^{N-1}$ and $T^{N-2} := \text{span}\{u_1, \dots, u_{N-2}\}$. Note that again $Px_{N-1} \neq 0$ holds almost surely. We repeat this procedure until we get a basis $\{u_{d+1}, \dots, u_N\}$ of P, which yields the desired space $L = P^\perp$. □

The previous theorem shows that, if we could compute the gradient in $(N - d)$ points, we would be able to recover the space L, respectively its orthogonal complement P, exactly. However, ATPC is not based on sampling the function f, since gradients cannot be computed exactly using point queries. Thus, we can only approximate the gradients by computing the divided differences

$$\nabla_h f(x) = \left[\frac{f(x + he_i) - f(x)}{h} \right]_{i=1}^N.$$

We adapt the first step in ATPC by substituting the computation of the gradients by computing divided difference and propose Algorithm 2, to which we will refer as *ATPE*, for the approximation task.

It then only remains to recover the ridge profile g. The estimation of g is rather straightforward. As the gradient gives the direction in which f changes, f becomes a one-dimensional function in the direction of the gradient. Hence, we can estimate g with well-known numerical methods. Indeed, we have already seen that the gradient of f in some point x is given by $\nabla f(x) = g'(\|Px\|_2^2)Px$, i.e., the normalized direction is $a := Px/\|Px\|_2$. Setting $x_t := ta$ yields

$$f(x_t) = g(\|Px_t\|_2^2) = g(\frac{t^2}{\|Px\|_2^2}\|Px\|_2^2) = g(t^2).$$

Algorithm 2: ATPE - Approximation by Adaptive Tangent Plane Estimation

Data: $f(\cdot) = g(\mathrm{dist}(\cdot, L)^2)$.
Input: $N \in \mathbb{N}$, $\dim(L) = d \in [N]$, $h, \hbar, \sigma > 0$.
Result: $\tilde{L}$.
begin
  $\tilde{f}^N \longleftarrow f$.
  $\tilde{T}^N \longleftarrow \mathbb{R}^N$.
  for $i = N, \dots, d+1$ **do**
    1. For some randomly chosen normalized $\tilde{x}_i \in \tilde{T}^i$, such that $\left|\tilde{f}^i(\tilde{x}_i) - f(0)\right| \geq \sigma$, compute

$$\nabla_h \tilde{f}^i(\tilde{x}_i) = \left[\frac{\tilde{f}^i(\tilde{x}_i + he_k) - \tilde{f}^i(\tilde{x}_i)}{h}\right]_{k=1}^{N}.$$

    2. $\tilde{u}_i \longleftarrow \nabla_h \tilde{f}^i(\tilde{x}_i)/\|\nabla_h \tilde{f}^i(\tilde{x}_i)\|_2$.
    3. $\tilde{T}^{i-1} \longleftarrow (\mathrm{span}\{\tilde{u}_i, \dots, \tilde{u}_N\})^{\perp}$.
    4. Let $\tilde{f}^{i-1}$ be the restriction of f to $\tilde{T}^{i-1}$.
  $\tilde{L} \longleftarrow \tilde{T}^d$.
  $\tilde{g} \longleftarrow \mathrm{interpolate}\{f(\sqrt{i\hbar}\tilde{u}_N)\}_{i=1}^m$, $m = 2\lfloor h^{-1} \rfloor$.
  $\tilde{f} \longleftarrow \tilde{g}(\mathrm{dist}(\cdot, \tilde{L})^2)$.

Note, that we need to estimate the gradients at random points $\tilde{x}_i \in \mathbb{R}^N$ such that $\left|\tilde{f}^i(\tilde{x}_i) - f(0)\right|$ is not too small in order to control the constants of the approximation. We therefore choose a threshold σ in advance, on which the approximation constants will depend. Thus in practice we choose a random sampling point and verify if this sampling point fulfills the requirement, if not we choose another random point. This might increase the number of necessary sampling points and the number will depend on the choice of the threshold σ. However, this number can be controlled. We will consider the following class of functions:

$$\begin{aligned}&\mathcal{LR}(s, d, c_2, c_3, \sigma, \varepsilon)\\ &:= \Bigg\{ f = g(\mathrm{dist}(\cdot, L))^2 \in \mathcal{LR}(s, d) : L \subset G(d, N),\, g'(t) \in [c_2, c_3] \text{ for all } t,\\ &\qquad \mathbb{P}\left(g(\sum_{i=1}^{N-d} x_i^2) \in [g(0) - \sigma, g(0) + \sigma]\right) \leq \varepsilon, x \in \mathbb{S}^{N-1} \text{ uniformly distributed}\Bigg\}\end{aligned}$$

The described procedure, of course, cannot find the correct plane P. However, it is able to compute a good approximation of P, where the approximation error depends on the choice of h.

For reasons of clarity, the proof of the next theorem is moved to the next subsection.

Theorem 3.2. *Let f be a linear-sleeve function of the form (4), i.e., $f \in \mathcal{LR}(s, d, c_2, c_3, \sigma, \varepsilon)$ for some $s \in (1, 2]$, $d \in [N]$, $\varepsilon \in (0, 1)$ and some positive constants c_2, c_3, σ. By sampling the function f at $(N-d)(N+1)$ appropriate points, ATPE constructs with probability larger than*

$(1-\varepsilon)^{N-d}$ *an approximation of* L *by a subspace* $\tilde{L} \subset \mathbb{R}^N$*, such that the error is bounded by*

$$\|L - \tilde{L}\|_{HS} \leq D\hat{C}(1+\hat{C}K)^{N-d}\sqrt{N-d}h^{s/2(s-1)^d},$$

for some arbitrarily small $h > 0$*, where*

$$\begin{aligned} K &= 4|g'|_{C^{s-1}} + 4c_3 \\ \hat{C} &= \sqrt{\frac{c_3}{c_2\sigma}} \\ D^s &= h^{2-s}c_3 + 4|g'|_{C^{s-1}}h^{s-1}(2+h)^{2(s-1)} + c_3|g'|_{C^{s-1}}(2+h)^{s-1}, \end{aligned}$$

which are constants depending only on the Hölder norm and bounds of g' *as well as on the choice of the thresholds* σ *and* ε*.*

In particular if g' *is Lipschitz continuous, i.e.* $s = 2$*, it holds*

$$\|L - \tilde{L}\|_{HS} \lesssim (1+\tilde{C})^{N-d}\sqrt{N-d}h,$$

with $\tilde{C} = \hat{C}K$*.*

If f *is defined on the unit* ℓ_2*-ball* $B_{\ell_2} = \{x \in \mathbb{R}^N : \|x\|_2 \leq 1\}$ *and* $0 < \hbar < 1$*, ATPE constructs an approximation* $\tilde{f}$ *of* f *such that the error is bounded by*

$$\|f - \tilde{f}\|_\infty \lesssim h^{s/2(s-1)^d} + \hbar.$$

Note, that similar to the algorithm in [16], this algorithm uses a fixed number of samples and the estimation cannot be improved by taking more samples. We therefore also aim for an algorithm which yields a reconstruction whose error decreases with the number of sampling points (cf. Section 4). Also note that due to the adaptive character of ATPE the reconstruction error increases for smaller values of the Lipschitz continuity s.

To complete this subsection, we want to remark that ATPE could be also adapted to the case that the dimension d of the subspace L is not known in advance. In that case, we would not stop the algorithm after $(N-d)$-steps, but if the norm of the gradient, or the divided differences, respectively, becomes smaller than a given threshold. Note that the restriction of f to T^d would be the constant function and thus the gradient would indeed vanish.

Finally, note that we can perform a similar, but slightly worse, error analysis for the case that f is of the form $f(x) = g(\|Px\|_2)$, whereas before we considered sleeve functions of the form $f(x) = g(\|Px\|_2^2)$. See Remark 1 in the next subsection for a short explanation.

3.2. Proof of Theorem 3.2

As mentioned above, the idea is to approximate the gradients of $f = f^N$ and f^i for $i = d+1, \ldots, N-1$. Since we need $N+1$ samples for each gradient approximation, we need $(N-d)(N+1)$ samples altogether. We already know from Theorem 3.1 that the subspace P can be written in terms of the gradients of the restrictions of f and f itself. Hence, we assume $P = L^\perp = \operatorname{span}\{u_{d+1} \ldots, u_N\}$, where the u_i's are given as stated in Theorem 3.1.

Note that we will prove that the constants in Theorem 3.2 are given by

$$\begin{aligned} K &= 4|g'|_{C^{s-1}} + 2c_3 + \max\{\|\nabla f(x_i)\|_2 : i = N, \ldots, d+1\} \\ \hat{C} &= 2/\min\{\|\nabla f^{N-i}(x_i)\|_2 : i = N, \ldots, d+1\} \\ D^s &= h^{2-s}c_3 + 4|g'|_{C^{s-1}}h^{s-1}(2+h)^{2(s-1)} + c_3|g'|_{C^{s-1}}(2+h)^{s-1}. \end{aligned}$$

The following estimations then yield the claim: By Eq. (6) we have

$$\|\nabla f(x)\|_2^2 = \|2g'(\|Px\|_2^2)Px\|_2^2 \leq 2c_3\|x\|_2^2 = 2c_3,$$

where we used that x is assumed to be normalized. Furthermore we derive

$$\left|g(\|Px\|_2^2) - g(0)\right| = \left|g'(\xi)\right| \|Px\|_2^2 \leq c_3\|Px\|_2^2,$$

which yields by assumption

$$\|Px\|_2^2 \geq \frac{\left|g(\|Px\|_2^2) - g(0)\right|}{c_3} = \frac{|f(x) - f(0)|}{c_3} \geq \frac{\sigma}{c_3},$$

with probability larger than $1 - \varepsilon$.

We split the proof by establishing several lemmata.

Lemma 3.3. *Under the assumptions of Theorem 3.2 and with the choice of the $\tilde{u}_i$'s, $i = d+1, \ldots, N$, as proposed in Algorithm 2, we have*

$$\|\tilde{u}_N - u_N\|_2 \leq \frac{2D\sqrt{N-d}h^{s/2}}{\|\nabla f(\tilde{x}_N)\|_2} =: S_0, \tag{7}$$

where $\tilde{x}_N \in \mathbb{R}^N$ is chosen randomly and $u_N = \frac{\nabla f(\tilde{x}_N)}{\|\nabla f(\tilde{x}_N)\|_2}$.

Proof. First, let us estimate the error between $\nabla f(x)$ and $\nabla_h f(x)$. We can compute

$$\begin{aligned}
\nabla_h f(x)_i &= \frac{f(x+he_i) - f(x)}{h} = \frac{g(\|P(x+he_i)\|_2^2) - g(\|Px\|_2^2)}{h} \\
&= g'(\xi_{i,h})\frac{\|P(x+he_i)\|_2^2 - \|Px\|_2^2}{h} \\
&= g'(\xi_{i,h})\frac{\sum_{j=d+1}^N \langle x+he_i, u_j\rangle^2 - \langle x, u_j\rangle^2}{h} \\
&= g'(\xi_{i,h})\sum_{j=d+1}^N 2u_{ji}\langle x, u_j\rangle + hu_{ji}^2 \\
&= g'(\xi_{i,h})\left(2[Px]_i + h\sum_{j=d+1}^N u_{ji}^2\right),
\end{aligned}$$

for some $\xi_{i,h}$ between $\|Px\|_2^2$ and $\|P(x+he_i)\|_2^2$, where u_{ji} denotes the ith entry of the vector u_j. We then estimate

$$\begin{aligned}
\left|\xi_{i,h} - \|Px\|_2^2\right| \leq \left|\|P(x+he_i)\|_2^2 - \|Px\|_2^2\right| &\leq \sum_{j=d+1}^N \left|\left(2h\langle x, u_j\rangle u_{ji} + h^2u_{ji}^2\right)\right| \\
&= 2h\left|[Px]_i\right| + h^2\sum_{j=d+1}^N u_{ji}^2 \leq 2h + h^2,
\end{aligned}$$

where we used the fact that u is a unit vector and that therefore all its entries (in absolute value) and the entries of its projection are smaller than or equal to one. Thus, the error which

we obtain by approximating the gradient can be estimated as

$$\begin{aligned}\|\nabla f(x) - \nabla_h f(x)\|_2^2 &\le \sum_{i=1}^N g'(\xi_{i,h})^2 h^2 \left(\sum_{j=d+1}^N u_{i,j}^2\right)^2 \\ &\quad + 4\sum_{i=1}^N \left(g'(\xi_{i,h}) - g'(\|Px\|_2)\right)^2 (Px)_i^2 \\ &\quad + 2h\sum_{i=1}^N \left|g'(\xi_{i,h})\right| \left|g'(\xi_{i,h}) - g'(\|Px\|_2^2)\right| |(Px)_i| \sum_{j=d+1}^N u_{ji}^2 \\ &=: T_1 + T_2 + T_3.\end{aligned}$$

To estimate those terms we take the following inequality into account:

$$\begin{aligned}\sum_{i=1}^N \left(\sum_{j=d+1}^N u_{ji}^2\right)^2 &= \sum_{i=1}^N \left(\sum_{j=d+1}^N u_{ji}^2\right)\left(\sum_{j=d+1}^N u_{ji}^2\right) \le \sum_{i=1}^N \left(\sum_{j=d+1}^N u_{ji}^2\right) \\ &= \sum_{j=d+1}^N \left(\sum_{i=1}^N u_{ji}^2\right) = N - d,\end{aligned}$$

where we used in the second as well as in the last step that $\{u_j\}_{j=1}^d$ forms an orthonormal system, which spans H, so that $\sum_{j=1}^d u_{ji}^2 \le 1$ for each $i = 1, \dots, N$ and $\sum_{i=1}^N u_{ji}^2 = 1$ for each $j = 1, \dots, d$. Now the desired estimates follow immediately:

$$\begin{aligned}T_1 &\le h^2 \|g'\|_\infty^2 \sum_{i=1}^N \left(\sum_{j=1}^d u_{ji}^2\right)^2 \le (N-d)\|g'\|_\infty^2 h^2, \\ T_2 &\le 4|g'|_{C^{s-1}}^2 (2h + h^2)^{2(s-1)} \sum_{i=1}^N (Px)_i^2 \le |g'|_{C^{s-1}}^2 (2h + h^2)^{2(s-1)}, \\ T_3 &\le \|g'\|_\infty \|g\|_{C^s} (N-d) h (2h + h^2)^{s-1}.\end{aligned}$$

Thus, we can find a constant $D > 0$, independent of the dimensions d and N, such that

$$\|\nabla f(x) - \nabla_h f(x)\|_2 \le D\sqrt{N-d}\, h^{s/2}. \tag{8}$$

The constant D can be chosen as stated by Theorem 3.2. Hence, applying Lemma 2.1, with $\lambda = 1/\|\nabla_h f(x)\|_2$, proves the claim. □

Next, we use the approximation of the gradient to approximate the tangent plane T^{N-1} at x with $\tilde{T}^{N-1} = \operatorname{span}\{\nabla_h f(x)^\perp\}$. The approximation error is then, of course, given by (7). Further, we let f^{N-1} and $\tilde{f}^{N-1}$ be the restriction of f to T^{N-1} and $\tilde{T}^{N-1}$, respectively.

Again we want to compute the column vectors u_i of V, $i = N, \dots, d+1$, step by step as the normalized gradients of f, f^i. But instead of computing the gradient of f^j we can only approximate it through an approximation of the gradient of $\tilde{f}^j$. Thus, we iteratively set the columns $\tilde{u}_i$, $i = N, \dots, d+1$ of $\tilde{V}$ as the normalized approximated gradients of $\tilde{f}_i$. The error of the approximation in each step can then be estimated by means of the following lemmata, in particular, by means of Lemma 3.4 for the first step and Lemma 3.7 for the ith step.

Before stating these lemmata, let us recall the definition of the matrices V, $\tilde{V}$, V_i and $\tilde{V}_i$, $i = N-1, \dots, d+1$. Let

$$\tilde{u}_i = \nabla_h \tilde{f}^i(\tilde{x}_i)/\|\nabla_h \tilde{f}^i(\tilde{x}_i)\|_2 \quad \text{and} \quad u_i = \nabla_h f^i(x_i)/\|\nabla_h f^i(x_i)\|_2,$$

for $i = N, \dots, d+1$, where $x_i = V V_i^T \tilde{x}_i$ and $\tilde{x}_i$ as well as f^i and $\tilde{f}^i$ are chosen as proposed in the Algorithms ATPC and ATPE. Then $\{u_{d+1}, \dots, u_N\}$ as well as $\{\tilde{u}_{d+1}, \dots, \tilde{u}_N\}$ form orthonormal systems and $u_i, \tilde{u}_i$, $i = 1, \dots, d$, are chosen that the whole systems $\{u_1, \dots, u_N\}$ and $\{\tilde{u}_1, \dots, \tilde{u}_N\}$ form an orthonormal basis for $\mathbb{R}^N$. We can now define

$$V_i = \begin{bmatrix} u_1 & u_2 & \dots & u_i & 0 & \dots & 0 \end{bmatrix}, \qquad V = \begin{bmatrix} u_1 & u_2 & \dots & u_N \end{bmatrix},$$
$$\tilde{V}_i = \begin{bmatrix} \tilde{u}_1 & \tilde{u}_2 & \dots & \tilde{u}_i & 0 & \dots & 0 \end{bmatrix}, \qquad \tilde{V} = \begin{bmatrix} \tilde{u}_1 & \tilde{u}_2 & \dots & \tilde{u}_N \end{bmatrix}.$$

Lemma 3.4. *With the same assumptions and choices as in Theorem 3.2 and Lemma 3.3, we have*

$$\|\tilde{u}_{N-1} - u_{N-1}\|_2 \leq \hat{C}(1+K) S_0^{s-1}.$$

We first have to prove the following lemma:

Lemma 3.5. *With the same assumptions and choices as in Theorem 3.2 and Lemma 3.3, let $x := V V_{N-1}^T \tilde{x}$, for some $\tilde{x} \in \tilde{T}^{N-1}$. We then have*

$$\|Px - P\tilde{x}\|_2 \leq \|\tilde{x}\|_2 \|u_N - \tilde{u}_N\|_2.$$

Proof. We write $\tilde{x} = \sum_{i=1}^{N-1} \tilde{x}_i \tilde{u}_i$ with $\tilde{x}_i \in \mathbb{R}$ and $|\tilde{x}_i| \leq 1$. Then we compute

$$\begin{aligned}
\|Px - P\tilde{x}\|_2^2 &= \| \sum_{i=d+1}^{N} \langle \sum_{j=1}^{N-1} \langle \tilde{x}, u_j \rangle u_j, u_i \rangle u_i - \langle \tilde{x}, u_i \rangle u_i \|_2^2 \\
&= \| \langle \tilde{x}, u_N \rangle u_N \|_2^2 = \| \langle \sum_{i=1}^{N-1} \tilde{x}_i \tilde{u}_i, u_N \rangle u_N \|_2^2 \\
&= \left(\sum_{i=1}^{N-1} \tilde{x}_i \langle \tilde{u}_i, u_N \rangle \right)^2 \overset{CS}{\leq} \sum_{i=1}^{N-1} \tilde{x}_i^2 \sum_{i=1}^{N-1} \langle \tilde{u}_i, u_N \rangle^2 \\
&= \left(\sum_{i=1}^{N-1} \tilde{x}_i^2 \right) \left(1 - \sum_{i=1}^{N} \langle \tilde{u}_i, u_N \rangle^2 + \sum_{i=1}^{N-1} \langle \tilde{u}_i, u_N \rangle^2 \right) \\
&\leq 2 \|\tilde{x}\|_2^2 \left(1 - \langle \tilde{u}_N, u_N \rangle \right).
\end{aligned}$$

In the step before the last we used that $\{\tilde{u}_1, \dots, \tilde{u}_N\}$ is an orthonormal basis (according to Step 2. and 3. in the ATPE Algorithm) and in the last step additionally that $1 - \langle \tilde{u}_N, u_N \rangle^2 = (1 + \langle \tilde{u}_N, u_N \rangle)(1 - \langle \tilde{u}_N, u_N \rangle) \leq 2(1 - \langle \tilde{u}_N, u_N \rangle)$. By observing that

$$\|u_N - \tilde{u}_N\|_2^2 = \langle u_N - \tilde{u}_N, u_N - \tilde{u}_N \rangle = \langle u_N, u_N \rangle - 2\langle u_N, \tilde{u}_N \rangle + \langle \tilde{u}_N, \tilde{u}_N \rangle = 2\left(1 - \langle u_N, \tilde{u}_N \rangle\right),$$

we deduce the claim. □

We are now able to prove the error of the first step of our algorithm ATPE, i.e., Lemma 3.4.

Proof of Lemma 3.4. For simplicity we will write $\tilde{x}$ and x instead of $\tilde{x}_{N-1}$ and x_{N-1}. Note that it is straightforward to show (compare to Eq. (6)) that $\nabla f^i(x) = V V_i^T \nabla f(x)$ and

$\nabla_h f^i(x) = \tilde{V}\tilde{V}_i \nabla_h f$. We hence obtain that

$$\begin{aligned}\|u_{N-1} - \tilde{u}_{N-1}\|_2 &= \left\| \frac{\nabla f^{N-1}(x)}{\|\nabla f^{N-1}(x)\|_2} - \frac{\nabla_h \tilde{f}^{N-1}(\tilde{x})}{\|\nabla_h \tilde{f}^{N-1}(\tilde{x})\|_2} \right\|_2 \leq \frac{2\|\nabla f^{N-1}(x) - \nabla_h \tilde{f}^{N-1}(\tilde{x})\|_2}{\|\nabla f^{N-1}(x)\|_2} \\ &\leq \frac{2}{\|\nabla f^{N-1}(x)\|_2} \Big(\|VV_{N-1}^T \nabla f(x) - \tilde{V}\tilde{V}_{N-1}^T \nabla f(x)\|_2 \\ &\quad + \|\tilde{V}\tilde{V}_{N-1}^T \nabla f(x) - \tilde{V}\tilde{V}_{N-1}^T \nabla f(\tilde{x})\|_2 + \|\tilde{V}\tilde{V}_{N-1}^T \nabla f(\tilde{x}) - \tilde{V}\tilde{V}_{N-1}^T \nabla_h f(\tilde{x})\|_2 \Big) \\ &=: \frac{2}{\|\nabla f^{N-1}(x)\|_2} (T_1 + T_2 + T_3),\end{aligned}$$

where we applied Lemma 2.1 with $\lambda = \|\nabla_h \tilde{f}^{N-1}(\tilde{x})\|_2$ in the second step. Now, we can estimate

$$\begin{aligned}T_1 &= \| \sum_{i=1}^{N-1} \langle u_i, \nabla f(x)\rangle u_i - \langle \tilde{u}_i, \nabla f(x)\rangle \tilde{u}_i\|_2 = \|\langle u_N, \nabla f(x)\rangle u_N - \langle \tilde{u}_N, \nabla f(x)\rangle \tilde{u}_N\|_2 \\ &\leq 2\|\nabla f(x)\|_2 \|u_N - \tilde{u}_N\|_2\end{aligned}$$

as well as

$$T_3 \leq \|\nabla f(\tilde{x}) - \nabla_h f(\tilde{x})\|_2 \leq 2D\sqrt{N-d}h^{s/2} = S_0.$$

Finally, we estimate T_2 by

$$\begin{aligned}T_2 &\leq \|\nabla f(x) - \nabla f(\tilde{x})\|_2 = 2\left\| g'(\|Px\|_2^2)Px - g'(\|P\tilde{x}\|_2^2)P\tilde{x} \right\|_2 \\ &\leq 2\Big[\left| g'(\|Px\|_2^2) - g'(\|P\tilde{x}\|_2^2) \right| \|Px\|_2 + \left| g'(\|P\tilde{x}\|_2^2) \right| \|Px - P\tilde{x}\|_2 \Big] \\ &\leq 2|g'|_{C^{s-1}} \left| \|Px\|_2^2 - \|P\tilde{x}\|_2^2 \right|^{s-1} + 2 \max_{|t|\leq 1} \{g'(t)\} \|Px - P\tilde{x}\|_2 \\ &\leq 4|g'|_{C^{s-1}} \|Px - P\tilde{x}\|_2^{s-1} + 2\|g\|_\infty \|Px - P\tilde{x}\|_2,\end{aligned}$$

which proves the lemma. □

One can now prove similar estimates as in Lemma 3.4 for $\|u_i - \tilde{u}_i\|_2$, $i = d+1, \ldots, N-2$. However, we first need to prove a more general version of Lemma 3.5.

Lemma 3.6. *With the same assumptions and choices as in Theorem 3.2 and Lemma 3.3, we have for $\tilde{x}_i \in \tilde{T}^i$ and $x = VV_i^T \tilde{x}_i$, $i = N-2, \ldots, d+1$, that*

$$\|Px_i - P\tilde{x}_i\|_2^2 \leq \|\tilde{x}_i\|_2^2 \sum_{j=i+1}^{N} \|\tilde{u}_j - u_j\|_2^2.$$

This inequality in turn yields the desired generalization of Lemma 3.4:

Lemma 3.7. *With the same assumption and choices as in Theorem 3.2 and Lemma 3.3, for $i = 0, \ldots, d$, we have*

$$\|u_{N-i} - \tilde{u}_{N-i}\|_2 \leq D(1 + K\hat{C})^i S_0^{(s-1)^i},$$

for all $i = 0, \ldots, N-d-1$.

Proof. Using the same methods as in the proof of Lemma 3.4 and using Lemma 3.6, one can show

$$\|u_{N-i} - \tilde{u}_{N-i}\|_2 \leq \hat{C} K (\sum_{j=0}^{i-1} S_j)^{s-1} + \hat{C} S_0,$$

where the S_j, $j = 1, \ldots, d$ are recursively defined. The claim then follows by induction. □

Putting the conclusions of the previous lemmata together and observing that

$$\sum_{i=0}^{N-d-1} (1 + K\hat{C})^i \leq (1 + K\hat{C})^{N-d}$$

finishes the proof of the first part of Theorem 3.2.

It remains to bound the error of the approximation of $\tilde{f}$. For that let $c := \|P\tilde{u}_N\|_2^2$ and $g_c = g(c\cdot)$ and remember that $\tilde{g} = \text{interpolate}\{f(\sqrt{i\hbar}\tilde{u}_N)\}_{i=1}^m = \text{interpolate}\{g(i\hbar\|P\tilde{u}_N\|_2^2)\}_{i=1}^m = \text{interpolate}\{g_c(i\hbar)\}_{i=1}^m$. Thus $\tilde{g}$ is a good approximation of g_c and therefore $\|\tilde{g} - g_c\|_\infty \lesssim \hbar$. The estimation of the error can now be divided in the following way:

$$\begin{aligned}\left|f(x) - \hat{f}(x)\right| &= \left|g(\|Px\|_2^2) - \widetilde{g}(\|\widetilde{P}x\|_2^2)\right| \\ &\leq \left|g(\|Px\|_2^2) - g_c(\|Px\|_2^2)\right| + \left|g_c(\|Px\|_2^2) - g_c(\|\widetilde{P}x\|_2^2)\right| \\ &\quad + \left|g_c(\|\widetilde{P}x\|_2^2) - \widetilde{g}(\|\widetilde{P}x\|_2^2)\right| = I + II + III.\end{aligned}$$

A finer estimation of I gives

$$\begin{aligned}\left|g(\|Px\|_2^2) - g(c\|Px\|_2^2)\right| &\leq (1-c)\|g'\|_\infty \|Px\|_2^2 \lesssim 1 - c = \|Pu_N\|_2^2 - \|P\tilde{u}_N\|_2^2 \\ &\leq 2\|P(u_N - \tilde{u}_N)\|_2^2 \\ &\leq 2\|u_N - \tilde{u}_N\|_2^2 \lesssim h^{s/2(s-1)^d}.\end{aligned}$$

II can be estimated analogously and clearly $III \lesssim \hbar$. This concludes the proof. □

As mentioned in the end of the last subsection, we can perform a similar, but slightly worse, error analysis for the case that f is of the form $f(x) = g(\text{dist}(x, L)) = g(\|Px\|_2)$. Indeed, we can estimate the approximation error of the gradient in the following way:

Remark 1. To approximate a function of the form $f(x) = g(\text{dist}(x, L)) = g(\|Px\|_2)$, for $x \in \mathbb{R}^N$, $g \in C^1([0,1])$ and g' Lipschitz continuous, we can utilize the following observation to obtain a worse approximation result than for linear sleeve functions of the form (4). Namely, we can rewrite the ith entry of the divided difference of $\|Px\|_2$ as:

$$\begin{aligned}\nabla_h(\|Px\|_2)_i &= \frac{\|P(x + he_i)\|_2 - \|Px\|_2}{h} = \frac{\|P(x+he_i)\|_2^2 - \|Px\|_2^2}{h\,(\|P(x+he_i)\|_2 + \|Px\|_2)} \\ &= \frac{2(Px)_i + h\|Pe_i\|_2^2}{\|P(x+he_i)\|_2 + \|Px\|_2}.\end{aligned}$$

We then obtain the following estimate:

$$\begin{aligned}&|\nabla(\|Px\|_2)_i - \nabla_h(\|Px\|_2)_i| \\ &= \left|\frac{2(Px)_i + h\|Pe_i\|_2^2}{\|P(x+he_i)\|_2 + \|Px\|_2} - \frac{(Px)_i}{\|Px\|_2}\right|\end{aligned}$$

$$\leq \frac{|2(Px)_i\|Px\|_2 - (Px)_i\|Px\|_2 - (Px)_i\|P(x+he_i)\|_2| + h\|Pe_i\|_2^2}{\|Px\|_2\,(\|Px\|_2 + \|P(x+he_i)\|_2)}$$
$$\leq \frac{|(Px)_i|\,\|Px - P(x+he_i)\|_2}{\|Px\|_2^2} + \frac{h\|Pe_i\|_2^2}{\|Px\|^2} \leq 2h\frac{\|Pe_i\|_2}{\|Px\|_2^2}.$$

Hence, we have:

$$\|\nabla(\|Px\|_2) - \nabla_h(\|Px\|_2)\|_2 \leq \frac{2h\sqrt{N-d}}{\|Px\|_2}.$$

If $\|Px\|_2$ is small, this upper bound can become large. Fortunately, in the case of linear sleeve functions of the form (4), we are not constrained by this term (cf. Eq. (8)).

4. An optimization algorithm on the Grassmannian manifold

We will now reformulate the given approximation problem as an optimization problem over the Grassmannian manifold $G(d, N)$. This reformulation allows us to develop an algorithm which yields a reconstruction whose error decreases with the number of sampling points. Remember that the previous algorithm needed a fixed number of sampling points and the error has decreased with the step size h in the computation of the divided differences. We again use the following notation for f

$$f(x) = g(\operatorname{dist}(x, L)^2) = g(\|Px\|_2^2),$$

where the operator P denotes the orthogonal projection P_P to the subspace $P \subset \mathbb{R}^N$ orthogonal to L. In the sequel of this section, we will, for brevity, assume that $\dim P = d$ where we assumed $\dim P = N - d = N - \dim L$ in the last section.

4.1. The algorithm

Let us assume without loss of generality that g is not the constant function. Otherwise we do not need to find the subspace, since every subspace can be used to represent g as stated. We first define for each $H \in G(d, N)$ a function f_H as linear-sleeve function, namely by

$$f_H(x) = g(\|Hx\|_2^2). \tag{9}$$

The main idea of the algorithm then uses the fact that $f_P = f$ and $f_H \neq f$ for $H \neq P$ and, thus, that P is the unique minimizer of

$$G(d, N) \ni H \mapsto \int_{[0,1]^N} |f(x) - f_H(x)|^2\,dx. \tag{10}$$

Note that we always mean the global minimizer of a function, if not differently referred. Unfortunately, we cannot express the objective function (10) in terms of sampling the input function f. On the one hand, we therefore need to replace the integral by a finite sum and on the other hand, the definition of f_H is not clear, since we do not know g in advance.

However, note that we can easily recover g by sampling f in some random direction $\hat{\theta}$. Indeed, it holds for $t \in \mathbb{R}$ that $f(t\hat{\theta}) = g(t\|P\hat{\theta}\|_2^2)$ and, since $\hat{\theta}$ is almost surely not contained in the orthogonal complement of P, g is up to the constant $\|P\hat{\theta}\|_2^2$ uniquely determined by $f(\cdot\hat{\theta})$. Hence, if we knew $\|P\hat{\theta}\|_2^2$ approximately, sampling f at $ih\hat{\theta}$, $i = 1, \ldots, M$, where $h \in (0, 1)$ is the step size, gave an approximation to g. Namely, with $g_{\hat{\theta}} = g(\cdot\|P\hat{\theta}\|_2^2)$,

we let $\hat{g}_{\hat{\theta}}^M$ be the approximation of $g_{\hat{\theta}}$ from the sampling points $(ih, f(ih\hat{\theta}))$. We then set $\hat{f}_H^M(x) = \hat{g}_\theta^M(\hat{p}_{\hat{\theta}}^{-1}\|Hx\|_2^2)$, where $\hat{p}_{\hat{\theta}}$ is an approximation of $\|P\hat{\theta}\|_2^2$.

One possibility to choose $\hat{\theta}$ such that we know $\|P\hat{\theta}\|_2^2$ approximately, is to choose $\hat{\theta}$ as the approximation of the normalized gradient of f in some random direction η. Indeed, the normalized gradient of f is given by

$$\theta = \frac{\nabla f(\eta)}{\|\nabla f(\eta)\|_2} = \frac{P\eta}{\|P\eta\|_2}.$$

Therefore, we have almost surely $\|P\theta\|_2 = 1$. Thus, we choose

$$\hat{\theta} = \frac{\tilde{\theta}}{\|\tilde{\theta}\|_2^2}, \qquad \text{where} \qquad \tilde{\theta}_i = \frac{f(\eta + he_i) - f(\eta)}{h}, \quad i = 1, \dots, N,$$

and let $\hat{g}_{\hat{\theta}}^M = Q_h g_{\hat{\theta}}$ be the approximation of $g_{\hat{\theta}}$, with Q_h as introduced in the preliminaries (see Eq. (5)).

The only remaining task is now to substitute the integral by a finite sum. Hence, we aim to define the objective function

$$\hat{F}^M : \quad G(d, N) \ni H \mapsto \sqrt{\sum_{i=1}^n \left| f(x_i) - \hat{f}_H^M(x_i) \right|^2}, \tag{11}$$

for some $x_1, \dots, x_n \in \mathbb{R}^N$ such that P is the unique minimizer of

$$F : \quad G(d, N) \ni H \mapsto \sqrt{\sum_{i=1}^n |f(x_i) - f_H(x_i)|^2}, \tag{12}$$

to ensure that the minimizer of $\hat{F}^M$ is a good approximation of P. Certainly, P can only be the unique minimizer of F if it uniquely minimizes the function

$$G(d, N) \ni H \mapsto \sqrt{\sum_{i=1}^n |\|Px_i\|_2 - \|Hx_i\|_2|^2}. \tag{13}$$

Therefore, it is necessary to find $x_1, \dots, x_n$ such that the map

$$A : G(d, N) \ni H \mapsto (\|Hx_1\|_2, \dots, \|Hx_n\|_2)$$

is injective. This problem is known as projection retrieval [12] and discussed in the next subsection.

The proposed procedure is summarized in Algorithm 3, to which we refer to as *OGM*.

We will now be able to prove the following main result in Section 4.3.

Theorem 4.1. *Let f be a linear-sleeve function of the form (4), i.e., $f \in \mathcal{LR}(s, d)$, $s \in (1, 2]$, $d \in [N]$, and $f = g(\mathrm{dist}(x, L)^2) = g(\|Px\|_2^2)$ for some d-dimensional subspace $P \subset \mathbb{R}^N$. Suppose that the derivative of g is bounded from below by some positive constant, and let $\hat{P} := \mathrm{argmin}_{H \in G(N-d,N)} \hat{\mathcal{F}}^M(H)$, where $\hat{\mathcal{F}}^M(H)$ is defined as in (14) with $\theta = \frac{\nabla f(\eta)}{\|\nabla f(\eta)\|_2}$ for some $\eta \in \mathbb{R}^N$. Then, we have*

$$\|\hat{P} - P\|_{HS} \lesssim M^{-s/2},$$

Algorithm 3: OGM - Approximation by Optimization over Grassmannian Manifold

For a given step size $h \in (0, 1)$ such that $M := h^{-1} \in \mathbb{N}$:

1. Choose direction $\theta \in \mathbb{S}^{N-1}$, such that we know $\|P\theta\|_2$ approximately (see explanations above).
2. Sample $y_i := f(ih\theta)$, $i = 1, \dots, M$.
3. Approximate $g_\theta := g(\cdot \|P\theta\|_2^2)$ by $\hat{g}_\theta^M = Q_h(g_\theta)$, with knots $\{(ih, y_i)\}_{i=1}^M$.
4. Approximate $\|P\theta\|_2$ by $\|P\hat{\theta}\|_2$.
5. Set $\hat{f}_H^M = \hat{g}_\theta^M(\frac{\|H\cdot\|_2^2}{\|P\hat{\theta}\|_2^2})$.
6. Minimize the objective function:

$$G(d, N) \ni H \mapsto \hat{\mathcal{F}}^M(H) = \sqrt{\sum_{i=1}^{n} \left| f(x_i) - \hat{f}_H^M(x_i) \right|^2}, \tag{14}$$

where $\hat{f}_H^M = \hat{g}_\theta^M(\frac{\|H\cdot\|_2^2}{\|P\hat{\theta}\|_2^2})$.

almost surely, with a constant depending only polynomially on the dimension of the space as well as on the derivatives of g and on $\|\nabla f(\eta)\|_2$. In particular, if $f \in \mathcal{LR}(2)$, then

$$\|\hat{P} - P\|_{HS} \lesssim M^{-1}.$$

Note that the statement holds indeed only almost surely, since we have to ensure that $P\eta \neq 0$.

Further note, that the objective function (14) is non-convex, since f and therefore $\hat{f}$ might be non-convex. Thus, to find the unique global minimizer might be very hard. Therefore, we shell run the minimization using an appropriate initial guess. This might be found by running ATPE.

The first step is to find measurements $\{x_i\}_{i=1}^n$ which ensure that the map defined in (13) is injective. Then we can turn to the error analysis and the proof of Theorem 4.1.

4.2. Projection retrieval

To find sampling points which ensure that the objective function has a unique minimizer, we consider the special case where g is the identity and $\mathcal{F}$ is therefore given by $\mathcal{F}(H) = \sum_{i=1}^n (\|Px_i\|_2 - \|Hx_i\|_2)^2$. Thus, P is the unique minimizer of $\mathcal{F}(H)$ if and only if the sampling points x_i, $i = 1, \dots, n$, determine P uniquely, i.e., if the map

$$A : G(d, N) \ni H \mapsto (\|Hx_1\|_2, \dots, \|Hx_n\|_2)$$

is injective. We can show the following theorem.

Theorem 4.2. *For every $P \in G(d, N)$ the quantities*

$$\|P(e_i + e_k)\|_2 =: \|Px_{i,k}\|_2, \qquad \text{for} \quad i = 1 \dots, N, \quad k = 1, \dots, N,$$

uniquely determine P.

Proof. Let $P, H \subset \mathbb{R}^N$ be two subspaces and let $\{u_1, \dots, u_d\}$ be an orthonormal basis for P and $\{v_1, \dots, v_d\}$ an orthonormal basis for H. Further, suppose that $\|Px_{i,k}\|_2^2 = \|Hx_{i,k}\|_2^2$ for $i = 1, \dots, N$, $k = i, \dots, N$. For $i = k$, we obtain

$$\sum_{j=1}^{d} v_{ji}^2 = \sum_{j=1}^{d} \langle e_i, v_j \rangle^2 = \|Hx_{ii}\|_2^2 = \|Px_{ii}\|_2^2 = \sum_{j=1}^{d} u_{ji}^2.$$

This shows that the entries of the diagonal of the projection matrices corresponding to P and H coincide. For the case $i \neq k$, we can compute

$$\sum_{j=1}^{d} v_{ji}^2 + 2\sum_{j=1}^{d} v_{ji}v_{jk} + \sum_{j=1}^{d} v_{jk}^2 = \|Hx_{ik}\|_2^2 = \|Px_{ik}\|_2^2 = \sum_{j=1}^{d} u_{ji}^2 + 2\sum_{j=1}^{d} u_{ji}u_{jk} + \sum_{j=1}^{d} u_{jk}^2.$$

Therefore, using the knowledge from the case where i equals k, this equation gives

$$\sum_{j=1}^{d} v_{ji}v_{jk} = \sum_{j=1}^{d} u_{ji}u_{jk}, \qquad \text{for} \quad i = 1, \dots, N, \quad k = i+1, \dots, N.$$

Thus, due to the symmetry of a real projection matrix, both projection matrices coincide, since the left-hand side equals the (i,k)th entry of the projection matrix corresponding to H and the right-hand side to P, respectively. Therefore, we can conclude that $H = P$. □

We see that we need $(N^2+N)/2$ sampling points to ensure injectivity, if we choose them as suggested by the last theorem. However, we believe a smaller number of sample points should be sufficient. Indeed, we can ensure that a fewer number of sampling points are sufficient to ensure almost surely injectivity, and therefore almost surely a unique minimizer. For this, we adapt Theorem 4 in [12] to the real case and deduce that to ensure almost injectivity, we require only $(d+1)(N+d/2)$ points.

Theorem 4.3 ([12]). *Let $d \leq N \in \mathbb{N}$. Draw a random subspace P uniformly with respect to the Haar measure from the Grassmannian manifold $G(d, N)$. Then the quantities*

$$\|Pe_i\|_2, \quad \|P(e_i + e_k)\|_2$$

for $i \in \{1, \dots, N\}$ and $k \in \{i+1, \dots, d\}$, uniquely determine P with a probability of 1.

Note that the change in the second index set in comparison to [12] is due to taking the symmetry of a real projection matrix into account. And the change in the number of necessary measurements is due to some small typo in [12], since for this proposed procedure we need to compute not only the first d columns of the projection matrix, but also all its diagonal entries. However, also in [12] it is proven that the first d columns of the projection matrix P determine the corresponding subspace almost surely uniquely. Thus, it would be desirable to find points which allow us to directly determine the entries of the first d columns, without computing all diagonal entries. This would deduce the number of necessary measurements to Nd. In the case $d = 1$, the following result by Fickus, Mixon, Nelson, Wang [13], tells us that $N+1$ measurements are sufficient to ensure almost injectivity, which is almost the conjectured number Nd.

Theorem 4.4 ([13]). *Consider $\Phi = \{\phi_i\}_{i=1}^n \subset \mathbb{R}^N$ and the intensity measurement mapping $A : \mathbb{R}^N/\pm 1 \to \mathbb{R}^n$ defined by $(A(x))(i) := |\langle x, \phi_i \rangle|^2$. Suppose each ϕ_i is nonzero. Then A is almost surely injective if and only if Φ spans $\mathbb{R}^N$ and rank Φ_S + rank $\Phi_{S^C} > N$ for each nonempty proper subset $S \subset \{1, \dots, N\}$.*

This shows that A cannot be almost injective if $n < N+1$. Moreover, for the case $n = N+1$, it is almost injective if and only if Φ is *full spark*, which means that every size-N collection of vectors of Φ is linearly independent. We remark that this result does not contradict the above mentioned conjecture that Nd measurements are sufficient. Indeed, in our case we want to recover the subspace and in this sense we can interpret the condition $1 = \|v\|_2 = \langle v, v\rangle$, for some vector $v \in \mathbb{R}^N$ of a orthonormal basis of this vector, as an additional measurement. Of course, it is not generally true that every size-N subcollection of $\{e_1, \dots, e_N, v\}$ forms a spanning set, because v could have zero entries. However, if we consider v as embedded in $\mathbb{R}^m$, where $\{i_1, \dots, i_m\}$ are the indices of the nonzero-entries of v, then every size-m subcollection of $\left\{e_{i_1}, \dots, e_{i_m}, v\right\}$ forms a spanning set for $\mathbb{R}^m$. Thus by Theorem 12 in [13], the entries $i_1, \dots, i_m$ of v are uniquely determined by these sampling points with a probability of 1. That the other entries are equal to zero is already determined by the other sampling points.

As indicated by this discussion, we suspect that Nd sampling points is sufficient to ensure almost surely injectivity of A. And, indeed, we are able to prove the following theorem, which states that even a fewer number of sampling points are sufficient, although, we do not directly determine the first d columns of the projection matrix.

Theorem 4.5. *Draw a random subspace P uniformly with respect to the Haar measure from the Grassmannian manifold $G(d, N)$. Then the quantities*

$$\|Pe_i\|_2^2, \quad \|P(e_j + e_k)\|_2^2, \quad \|Px\|_2^2$$

for a randomly chosen vector $x \in \mathbb{R}^N$, $i \in \{1, \dots, N-1\}$, $j \in \{1, \dots, d\}$ and $k \in \{j+1, \dots, N\}$, uniquely determine P with a probability of 1.

Proof. We start with the same argumentation as in [12], which tells us that the first d columns of the projection matrix P are linearly independent for almost every P, that the diagonal entries of the projection matrix P are given by

$$P_{ii} = \|Pe_i\|_2^2, \quad \text{for} \quad i = 1, \dots, N,$$

and that the other entries can be computed as

$$2P_{ij} = \|P(e_i + e_j)\|_2^2 - \|Pe_i\|_2^2 - \|Pe_j\|_2^2.$$

We further note that we only need to observe $N-1$ diagonal entries to determine all N diagonal entries of P. Indeed, if $u_1, \dots, u_d$ is any orthonormal system which spans P, it holds that

$$\sum_{i=1}^{N} P_{ii} = \sum_{i=1}^{N} \|Pe_i\|_2^2 = \sum_{i=1}^{N} \sum_{j=1}^{d} u_{ji}^2 = d,$$

and, thus, $P_{NN} = d - \sum_{i=1}^{N-1} P_{ii}$. Hence, by observing $\|Pe_i\|_2^2$, $i = 1, \dots, N-1$, as well as $\|P(e_j + e_k)\|_2^2$, $j = 1, \dots, d$ and $k = j+1, \dots, N$, we can recover the first $d-1$ columns of the projection matrix P.

We now claim that there exist only finitely many projection matrices with the same first $d-1$ columns. This would show that there are only finitely many subspaces H which yield the same measurements as P for the stated collection of quantities and that we therefore can almost surely uniquely recover P by an additional random measurement.

Thus, suppose we already know the first $d-1$ columns of the projection matrix. These are clearly linearly independent considering the same random event, in which the first d columns

of P are linearly independent. We can now use the fact that for a projection matrix P, it has to hold that $PP = P$, and hence that each column of P lies in the span of the corresponding subspace. Applying Gram–Schmidt orthonormalization to the first $d-1$ columns, which are linearly independent, gives an orthonormal system of $d-1$ vectors, which we denote by $u_1, \dots, u_{d-1}$. Thus, there is only one unknown basis vector, denoted by u_d, for the subspace P left. However, the measurements we have already taken determine the entries of this vector uniquely in absolute value. Indeed, we have

$$\|Pe_i\|_2^2 = \sum_{j=1}^{d} u_{ji}^2 = u_{di}^2 + \sum_{j=1}^{d-1} u_{ji}^2,$$

which is equivalent to

$$u_{di}^2 = \|Pe_i\|_2^2 - \sum_{j=1}^{d-1} u_{ji}^2.$$

Note that we already know the right-hand side of the last equation. This shows that there are indeed only finitely many subspaces which produce the same measurements as P. Hence, taking some random measurement $\|Px\|_2^2$ for some randomly chosen $x \in \mathbb{R}^N$ in addition, yields the desired almost injectivity. □

The following corollary shows that if the dimension of the subspace is $d > N/2$, we can choose the same measurements as if the dimension would be $N-d$. So we can further deduce the number of measurements. For example, if the dimension $d = N-1$, we can apply Theorem 4.4 and find that N measurements are sufficient to ensure almost injectivity of A.

Corollary 4.6. *With the same choice of measurements as in Theorem 4.5 we can uniquely determine a randomly drawn subspace $P \in G(N-d, N)$ with a probability of 1, i.e., the measurements*

$$\|Pe_i\|_2^2, \quad \|P(e_i + e_k)\|_2^2, \quad \|Px\|_2^2$$

for some random vector $x \in \mathbb{R}^N$, $i \in \{1, \dots, N-1\}$, $j \in \{1, \dots, d\}$ and $k \in \{j+1, \dots, N\}$, uniquely determine P with probability 1.

Proof. For every $y \in \mathbb{R}^N$ it holds $\|Py\|_2^2 = \|Hy\|_2^2$ if and only if $\|P^\perp y\|_2^2 = \|y\|_2^2 - \|Py\|_2^2 = \|y\|_2^2 - \|Hy\|_2^2 = \|H^\perp y\|_2^2$, and therefore, we can apply the results of the above theorem. □

4.3. Proof of Theorem 4.1

In the last subsection we have shown that we need less than Nd sampling points to ensure that the objective function defined in (13), where g is assumed to be the identity, has almost surely a unique minimizer. However, for ease of computation we will use the sampling points proposed in Theorem 4.2. We first show that the measurements given in Theorem 4.2 also ensure a unique minimizer of F, which was defined in (12). For this purpose we introduce a bijective mapping

$$\iota : \{1, \dots, N(N+1)/2\} \to \{(j,k) : j \in \{1, \dots, N\}, k \in \{j, \dots, N\}\}$$

and set $n = N(N+1)/2$ as well as

$$x_i := x_{\iota(i)} = x_{j,k} = \begin{cases} e_j & \text{if } j = k, \\ e_j + e_k & \text{if } j \neq k, \end{cases} \qquad \text{for } i = 1, \dots, n.$$

Lemma 4.7. *Suppose that g fulfills the assumption of Theorem 4.1. Then P is almost surely the unique minimizer of*

$$\mathcal{F} : G(d,N) \to \mathbb{R}, \quad H \mapsto \sqrt{\sum_{i=1}^{n} (f(x_i) - f_H(x_i))^2},$$

where x_i, $i = 1, \dots, n$, are defined as above and f_H in (9).

Proof. Suppose $H \in G(d,N)$ is another minimizer. Then for all $i \in \{1, \dots, n\}$, we conclude $g(\|Px_i\|_2^2) = g(\|Hx_i\|_2^2)$. But since g is injective, this implies $\|Px_i\|_2 = \|Hx_i\|_2$ for all $i \in \{1, \dots, n\}$. Thus by the statements proved in Section 4.2 we conclude that $P = H$. □

Lemma 4.8. *Under the same assumption as in Theorem 4.1, we have*

$$|\hat{\mathcal{F}}^M(H) - \mathcal{F}(H)| \leq C\sqrt{d}M^{-s/2},$$

where $\hat{\mathcal{F}}^M$ was introduced in Eq. (11) and the constant depends on the derivatives of g and on $\|\nabla f(\eta)\|_2$.

Proof. We start by estimating $\hat{\mathcal{F}}^M(H)$ as

$$\begin{aligned} \hat{\mathcal{F}}^M(H) &= \sqrt{\sum_{i=1}^{n} \left(f(x_i) - \hat{f}_H^M(x_i) \right)^2} \\ &= \sqrt{\sum_{i=1}^{n} \left((f(x_i) - f_H(x_i)) + (f_H(x_i) - \hat{f}_H^M(x_i)) \right)^2} \\ &\leq \sqrt{\sum_{i=1}^{n} (f(x_i) - f_H(x_i))^2} + \sqrt{\sum_{i=1}^{n} (f_H(x_i) - \hat{f}_H^M(x_i))^2} \\ &= \mathcal{F}(H) + \sqrt{\sum_{i=1}^{n} (f_H(x_i) - \hat{f}_H^M(x_i))^2}, \end{aligned}$$

where we used the triangle inequality for $\|\cdot\|_{\ell_2^n}$ in the second step. We can apply a similar argument to $\mathcal{F}(H)$ to derive $\mathcal{F}(H) \leq \hat{\mathcal{F}}^M(H) + \sqrt{\sum_{i=1}^{n} (f_H(x_i) - \hat{f}_H^M(x_i))^2}$. This in turn yields the inequality

$$|\mathcal{F}(H) - \hat{\mathcal{F}}^M(H)| \leq \sqrt{\sum_{i=1}^{n} (f_H(x_i) - \hat{f}_H^M(x_i))^2}.$$

We now split this inequality by

$$\begin{aligned}
&|\mathcal{F}(H) - \hat{\mathcal{F}}^M(H)| \\
&\leq \sqrt{\sum_{i=1}^{n} \left(g(\|Hx_i\|_2^2) - g_{\hat{\theta}}(\|Hx_i\|_2^2) + g_{\hat{\theta}}(\|Hx_i\|_2^2) - \hat{g}_{\hat{\theta}}^M(\|Hx_i\|_2^2) \right)^2} \\
&\leq \sqrt{\sum_{i=1}^{n} \left(g(\|Hx_i\|_2^2) - g_{\hat{\theta}}(\|Hx_i\|_2^2) \right)^2} + \sqrt{\sum_{i=1}^{n} \left(g_{\hat{\theta}}(\|Hx_i\|_2^2) - \hat{g}_{\hat{\theta}}^M(\|Hx_i\|_2^2) \right)^2} \\
&\leq n \left(\|g - g_{\hat{\theta}}\|_\infty + \|g_{\hat{\theta}} - \hat{g}_{\hat{\theta}}^M\|_\infty \right) =: n(T_1 + T_2).
\end{aligned}$$

For T_1, we estimate

$$\begin{aligned}
T_1 &= \sup_{t\in[0,1]} \left| g(t) - g(\|P\hat{\theta}\|_2^2 t) \right| \leq \|g'\|_\infty |1 - \|P\hat{\theta}\|_2^2| \\
&= \|g'\|_\infty \left| \|P \frac{\nabla f(\eta)}{\|\nabla f(\eta)\|_2}\|_2^2 - \|P \frac{\nabla_h f(\eta)}{\|\nabla_h f(\eta)\|_2}\|_2^2 \right| \\
&\leq 2\|g'\|_\infty \|\frac{\nabla f(\eta)}{\|\nabla f(\eta)\|_2} - \frac{\nabla_h f(\eta)}{\|\nabla_h f(\eta)\|_2}\|_2 \leq 4\|g'\|_\infty \frac{\|\nabla f(\eta) - \nabla_h f(\eta)\|_2}{\|\nabla f(\eta)\|_2} \\
&\lesssim \sqrt{d} h^{s/2},
\end{aligned}$$

with a constant depending on $\|g'\|_\infty$ and $\|\nabla f(\eta)\|_2$. Note that we used the estimate from Lemma 3.3.

Using Property (5), we can bound T_2 by

$$T_2 = \sup_{t\in[0,1]} \left| g_{\hat{\theta}}(t) - g_{\hat{\theta}}^M(t) \right| \leq C h^s,$$

where C is a constant depending only on the degree of the interpolating polynomials. This proves the lemma. □

Theorem 4.9 (*Convergence*). *Under the assumptions of Theorem 4.1, suppose that $\hat{P}$ is a minimizer of $\hat{\mathcal{F}}^M$. Then it almost surely holds*

$$\|P - \hat{P}\|_{HS} \lesssim n(N+1)\sqrt{d} M^{-s/2},$$

with a constant depending on the Hölder norm and bounds of g' as well as on $\|\nabla f(\eta)\|_2$.

Proof. Let H_0 be a minimizer of $\hat{\mathcal{F}}^M$. First note that

$$\mathcal{F}(H_0) \leq 2C\sqrt{d} h^{s/2},$$

because $\hat{\mathcal{F}}^M(H_0) \leq \hat{F}^M(P) = \hat{\mathcal{F}}^M(P) - \mathcal{F}(P) \leq C\sqrt{d} h^{s/2}$, where we used the fact that H is a minimizer in the first step, that $\mathcal{F}(P) = 0$ in the second step and the statement of the last lemma in the third step. The stated bound then follows from $\mathcal{F}(H_0) \leq \left| \mathcal{F}(H_0) - \hat{\mathcal{F}}^M(H_0) \right| +$

$\left|\hat{\mathcal{F}}^M(H_0)\right| \leq 2C\sqrt{d}h^{s/2}$. This yields

$$\begin{aligned} 2C\sqrt{d}M^{-s/2} \geq \mathcal{F}(H) &= \sqrt{\sum_{i=1}^{n}\left(g(\|Px_i\|_2^2) - g(\|Hx_i\|_2^2)\right)^2} \\ &\geq \min|g'|\sqrt{\sum_{i=1}^{n}\left(\|Px_i\|_2^2 - \|Hx_i\|_2^2\right)^2}. \end{aligned} \tag{15}$$

Now define the matrix $\tilde{P}$ by

$$\tilde{P}_{ii} = \|Pe_i\|_2^2 \quad \text{and} \quad \tilde{P}_{ij} = \|P(e_i + e_j)\|_2^2,$$

and $\tilde{H}$ analogously by

$$\tilde{H}_{ii} = \|H_0 e_i\|_2^2 \quad \text{and} \quad \tilde{H}_{ij} = \|H_0(e_i + e_j)\|_2^2.$$

We further denote the matrix which only contains the diagonal entries of a matrix P, and is zero elsewhere, by P_D and the matrix which is zero along the diagonal and coincides at all off-diagonal entries with P by P_{OD}. We then have

$$\sqrt{\sum_{i=1}^{n}\left(\|Px_i\|_2^2 - \|H_0x_i\|_2^2\right)^2} = \|\tilde{P}_D - \tilde{H}_D + \frac{1}{\sqrt{2}}\left(\tilde{P}_{OD} - \tilde{H}_{OD}\right)\|_{\mathrm{F}}.$$

Note that the $\{ij\}$th entry P_{ij} of the projection matrix P to the belonging subspace is given by

$$P_{ii} = \|Pe_i\|_2^2 \quad \text{and} \quad 2P_{ij} = \|P(e_i + e_j)\|_2^2 - \|Pe_i\|_2^2 - \|Pe_j\|_2^2.$$

Thus, defining

$$B = \begin{bmatrix} 0 & P_{11} - H_{11} & \dots & P_{11} - H_{11} \\ P_{22} - H_{22} & 0 & \dots & P_{22} - H_{22} \\ \vdots & & & \vdots \\ P_{NN} - H_{NN} & \dots & P_{NN} - H_{NN} & 0 \end{bmatrix},$$

gives

$$\begin{aligned} \|P - H\|_{\mathrm{F}} &= \left\|\tilde{P}_D - \tilde{H}_D + \frac{1}{2}\left(\tilde{P}_{OD} - \tilde{H}_{OD} - B - B^T\right)\right\|_{\mathrm{F}} \\ &\leq \left\|\tilde{P}_D - \tilde{H}_D + \frac{1}{\sqrt{2}}\left(\tilde{P}_{OD} - \tilde{H}_{OD}\right)\right\|_{\mathrm{F}} + \frac{\sqrt{2}-1}{2}\|\tilde{P}_{OD} - \tilde{H}_{OD}\|_{\mathrm{F}} + \|B\|_{\mathrm{F}} \\ &\leq \sqrt{\sum_{i=1}^{n}\left(\|Px_{ij}\|_2^2 - \|Hx_{ij}\|_2^2\right)^2} + (N-1)\|\tilde{P}_D - \tilde{H}_D\|_{\mathrm{F}} + 1/4\|(\tilde{P})_{OD} - \tilde{H}_{OD}\|_{\mathrm{F}} \\ &\leq (N+1)\sqrt{\sum_{i=1}^{n}\left(\|Px_{ij}\|_2^2 - \|Hx_{ij}\|_2^2\right)^2}. \end{aligned}$$

Applying (15), we can therefore deduce

$$\frac{\min|g'|}{N+1}\|P-H\|_{\mathrm{F}} \leq \min|g'| \sqrt{\sum_{i=1}^{n}\left(\|Px_i\|_2^2 - \|Hx_i\|_2^2\right)^2} \leq 2C\sqrt{d}M^{-s/2},$$

which proves the claim. □

5. Numerical results

In this section we investigate the performance of the approximation schemes presented in the last sections. Our algorithms separate the approximation task in approximating the one-dimensional function g and the subspace P independently. Consequently, the quality of the uniform approximation of f by $\hat{f}$ is then bounded by the corresponding error between g and $\hat{g}$ and the error between P and $\hat{P}$. In what follows, we will only discuss the approximation error between P and $\hat{P}$, because the approximation error between g and $\hat{g}$ is well known, cf. [10] and Section 2.

We consider two different functions, one which fulfills all assumptions of Theorem 4.1, namely $g = \tanh$, and one which does not fulfill the assumption of a positive derivative, namely $g = \sin(5\cdot)$, which is not monotone on its domain. We further consider for the dimension of the subspace $d = 1$ and 8. For the dimension of the ambient space we choose $N = 10$ and 50. For each combination of N, d and g we ran 100 experiments, where we drew a subspace $P \in G(d, N)$ uniformly at random. Note that the analysis of the algorithm makes heavy use of the monotonicity of g. However, the numerics show that we have comparably good results for the non-monotonic function $\sin(5\cdot)$.

5.1. Numerical results for ATPE

The implementation of ATPE (Algorithm 2) is straightforward. However, to draw a random vector from some tangent plane, we used a method of the Matlab toolbox Manopt [1] to draw a random vector of $\mathbb{R}^N$ and projected it to the tangent plane. The results can be seen in Fig. 4. They show as promised that the error of the approximation becomes arbitrarily small for all considered choices of d, N and g, if we choose h in computing the divided differences certainly small.

5.2. Numerical Results for OGM (Algorithm 3)

To solve the optimization problem (14), we leverage the freely available Matlab toolbox Manopt [1]. In particular, we imposed the manifold constraint by choosing the built-in `grassmann- factory` and we selected the built-in `steepestdescent` solver, as steepest descent is a well-known method to solve optimization problems. This solver requires both a cost function and the Euclidean gradient of the cost function as inputs:

$$\begin{aligned}\texttt{cost(H)} &= 0.25\sum_{k=1}^{n}\left(f(x_k) - \texttt{interp1}(\{ih\}_{i=1}^{N}, \{f(\|P(ih\eta)\|_2^2)\}_{i=1}^{M}, \|Hx_k\|_2^2, \text{`spline'})\right)^2 \\ &=: 0.25\sum_{k=1}^{n}\left(f(x_k) - \mathbf{f}(x_k)\right)\end{aligned}$$

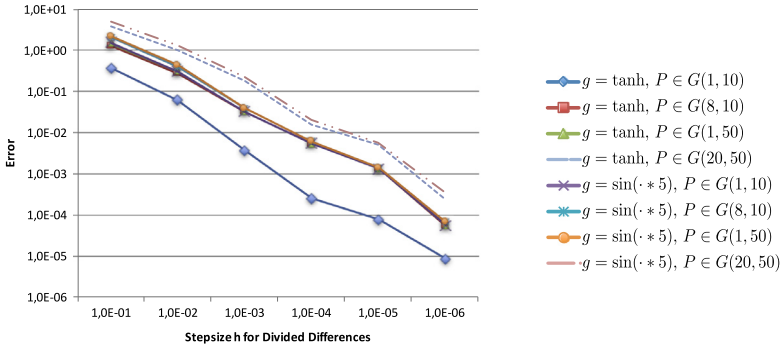


Fig. 4. Average error of the approximation of randomly drawn subspaces P using ATPE depending of the step size h in computing $\nabla_h f$. We plotted the error as the Hilbert–Schmidt distance $\|P - \tilde{P}\|_{\mathrm{HS}}$ between the groundtruth P and the reconstruction $\tilde{P}$.

$$\texttt{egrad} = \sum_{k=1}^{n} (f(x_k) - \mathbf{f}(x_k))\,(\texttt{interp1}(\{ih\}_{i=1}^{N}, \{f(\|P(ih\eta)\|_2^2)\}_{i=1}^{M},$$
$$\|Hx_k\|_2^2 + h/100, \text{‘spline’}) - \mathrm{f}(x_k))\mathbf{H}_k,$$

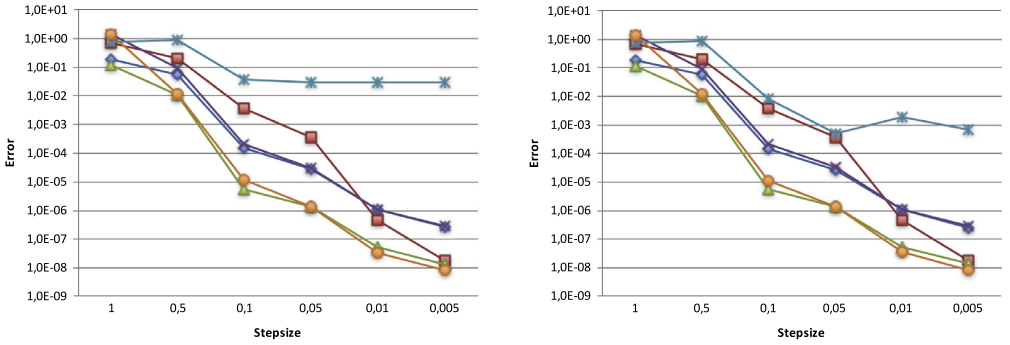
where $\mathbf{H}_k = [H_1 x_k x_k \dots H_N x_k x_k]$ and $\texttt{interp1}(x, v, xq, \text{‘spline’})$ return interpolated values of a one-dimensional function at specific query points xq using spline interpolation. The vector x contains the sample points, and v contains the corresponding function values. Note, that the Euclidean gradient ignores the manifold constraints.

Further observe, that even so OGM is shown to succeed to find an objective function whose minimizer is a suitable approximation of the wanted subspace, it is not obvious that the optimization algorithm can succeed to find this minimizer. In order to hope for this, we need to input a default subspace to the optimization algorithm which is not too far from the wanted one. In the following we choose those default subspaces uniformly at random in two different neighborhoods of the wanted subspace.

Fig. 5 shows the results for both functions and the different choices of N and d, if the default value for the optimization is a randomly chosen subspace in a distance of at most $\sqrt{2d(1 - \cos(\pi/3))}$ to the unique minimum P of the objective function (11), i.e., if the default value is a rotation of P with an angle of at most $\pi/3$. The error is given in a logarithmic scale and the lines correspond to the different choices of g, N and d. We see that we can recover all randomly drawn subspaces successfully, whenever the dimension is $d = 1$ or whenever $g = \tanh$. This fits to our analysis, where the theorems hold true for injective functions and indeed $\sin(5\cdot)$ is not at all injective.

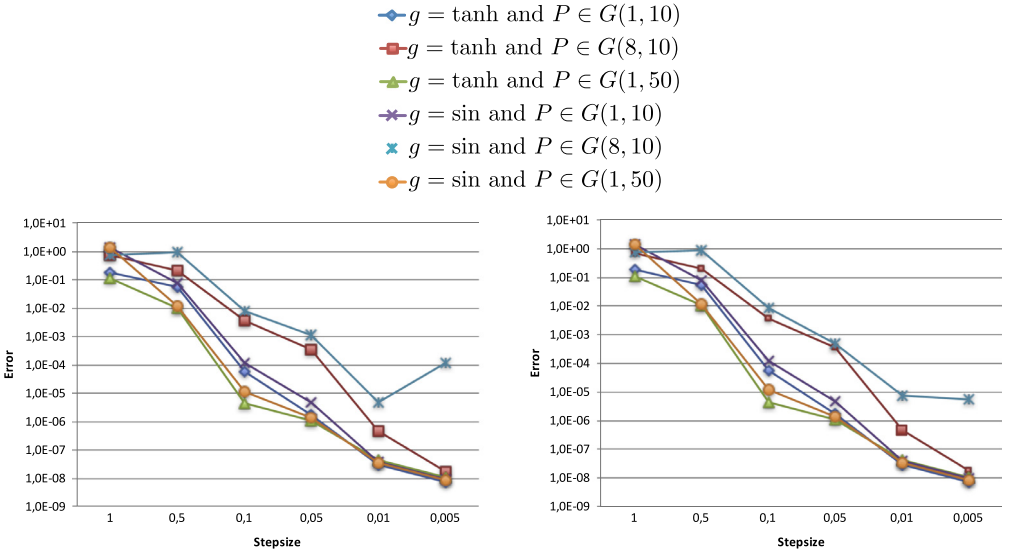
Fig. 6 shows that for the case that the dimension is $d = 8$ and that we have $g = \sin(5\cdot)$, we can recover 95% of randomly drawn subspaces if we ensure that the default value for the optimization is at most in a distance of $\sqrt{2d(1 - \cos(\pi/4))}$ to P. Thus, we see that for a more carefully chosen default value, all subspaces can be recovered with a reasonable small error. Note that the case $d = 1$ corresponds to the case of a usual ridge function, because if $\dim P = 1$ we measure the distance to the $N - 1$-dimensional subspace $P^{\perp}$.

Furthermore, as we have seen in Section 4.2 we can use fewer measurements to ensure almost injectivity. We then also have to adapt the convergence analysis of $\hat{F}_M$.



(a) Average (b) 95% Quantile

Fig. 5. Default value for optimization is a random rotation of P by a factor of at most $\pi/3$. We plotted the error as the Hilbert–Schmidt distance $\|P - \tilde{P}\|_{\mathrm{HS}}$ between the groundtruth P and the reconstruction $\tilde{P}$.



(a) Average (b) 95% Quantile

Fig. 6. Default value for optimization is a random rotation of P by a factor of at most $\pi/4$. We plotted the error as the Hilbert–Schmidt distance $\|P - \tilde{P}\|_{\mathrm{HS}}$ between the groundtruth P and the reconstruction $\tilde{P}$.

Acknowledgments

The author acknowledges support by the DFG Grant 1446/18 and the Berlin Mathematical School. In particular the author acknowledges Ingrid Daubechies, Gitta Kutyniok and Mauro Maggioni for helpful discussions.

References

[1] P.-A. Absil, N. Boumal, B. Mishra, R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, J. Mach. Learn. Res. 15 (2014) 1455–1459.

[2] W.K. Allard, G. Chen, M. Maggioni, Multi-scale geometric methods for data sets II: Geometric multi-resolution analysis, Appl. Comput. Harmon. Anal. 32 (3) (2012) 435–462.

[3] R.E. Bellman, Adaptive Control Processes: a Guided Tour, Princeton University Press, 1961.

[4] E.J. Candès, Harmonic analysis of neural networks, Appl. Comput. Harmon. Anal. 6 (2) (1999) 197–218.

[5] E.J. Candès, Y. Plan, Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements, IEEE Trans. Inform. Theory 57 (4) (2011) 2342–2359.

[6] A. Cohen, I. Daubechies, R.A. DeVore, G. Kerkyarcharian, D. Picard, Capturing ridge functions in high dimensions from point queries, Constr. Approx. 35 (2012) 225–243.

[7] A. Cohen, R.A. DeVore, C. Schwab, Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs, Found. Comput. Math. 10 (6) (2010) 615–646.

[8] M. Davenport, C. Hegde, M.F. Duarte, R.G. Baraniuk, Joint manifolds for data fusion, IEEE Trans. Image Process. 19 (10) (2010) 2580–2594.

[9] R.R. DeVore, Nonlinear approximation, Acta Numer. 7 (1998) 51–150.

[10] R.A. DeVore, G.G. Lorentz, Constructive Approximation, Vol. 303, Springer, 1993.

[11] R.A. DeVore, G. Petrova, P. Wojtaszczyk, Approximation of functions of few variables in high dimensions, Constr. Approx. 33 (1) (2011) 125–143.

[12] M. Fickus, D.G. Mixon, Projection retrieval: Theory and algorithms, in: 2015 International Conference on Sampling Theory and Applications, SampTA.

[13] M. Fickus, D.G. Mixon, A.A. Nelson, Y. Wang, Phase retrieval from very few measurements, Linear Algebra Appl. 449 (2014) 475–499.

[14] M. Fornasier, K. Schnass, J. Vybiral, Learning functions of few arbitrary linear parameters in high dimensions, Found. Comput. Math. 12 (2012) 229–262.

[15] S. Keiper, Analysis of Generalized High-Dimensional Ridge Functions (Masters thesis), TU Berlin, 2015.

[16] A. Kolleck, J. Vybiral, On some aspects of approximation of ridge functions, J. Approx. Theory 194 (2015) 35–61.

[17] M.H. Maathuis, M. Kalisch, P. Bühlmann, Estimating high-dimensional intervention effects from observational data, Ann. Statist. 37 (6A) (2009) 3133–3164.

[18] S. Mayer, T. Ullrich, J. Vybíral, Entropy and sampling numbers of classes of ridge functions, Constr. Approx. (2014) 1–34.

[19] E. Novak, H. Woźniakowski, Approximation of infinitely differentiable multivariate functions is intractable, J. Complexity 25 (4) (2009) 398–404.

[20] H. Tyagi, V. Cevher, Active learning of multi-index function models, in: Adv. Neural Inf. Process. Syst., 2012, pp. 1466–1474.

[21] H. Tyagi, V. Cevher, Learning ridge functions with randomized sampling in high dimensions, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2012, pp. 2025–2028.

[22] M.J. Wainwright, Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting, IEEE Trans. Inform. Theory 55 (12) (2009) 5728–5741.