SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 7, No. 4, pp. 1301–1322

Dimension Reduction via Gaussian Ridge Functions*

Pranay Seshadri[†], Shaowu Yuchi[‡], and Geoffrey T. Parks[§]

Abstract. Ridge functions have recently emerged as a powerful set of ideas for subspace-based dimension reduction. In this paper we begin by drawing parallels between ridge subspaces, sufficient dimension reduction and active subspaces, contrasting between techniques rooted in statistical regression and those rooted in approximation theory. This sets the stage for our new algorithm that approximates what we call a Gaussian ridge function—the posterior mean of a Gaussian process on a dimension-reducing subspace—suitable for both regression and approximation problems. To compute this subspace we develop an iterative algorithm that alternates between optimizing over the Stiefel manifold to compute the subspace and optimizing the hyperparameters of the Gaussian process. We demonstrate the utility of the algorithm on two analytical functions, where we obtain near exact ridge recovery, and a turbomachinery case study, where we compare the efficacy of our approach with three well-known sufficient dimension reduction methods: SIR, SAVE, and CR. The comparisons motivate the use of the posterior variance as a heuristic for identifying the suitability of a dimension-reducing subspace.

Key words. Gaussian process regression, dimension reduction, manifold optimization, active subspaces, sufficient dimension reduction

AMS subject classifications. 60G15, 51M35

DOI. 10.1137/18M1168571

1. Introduction.

Dimension reduction is essential for modern data analysis. Its utility spans both simulation informatics and statistical regression fields. The objective of dimension reduction is to obtain parsimoniously parameterized models—functions of few variables that can be used for predicting, understanding, and visualizing the trends observed in high-dimensional data sets. These data sets may originate from a tailored *design of experiment* of computer models (see [30]) and are therefore deterministic, or they may originate from the observations of sensors and are therefore subject to measurement noise. Across both paradigms, techniques and ideas within *subspace-based dimension reduction* have proven to be extremely fruitful in inference. Application areas that have benefitted from subspace-based

[*] Received by the editors February 2, 2018; accepted for publication (in revised form) July 26, 2019; published electronically November 7, 2019. This paper was initially submitted to a special section proposed on "Approaches to Reduce the Dimension of Model Parameters." JUQ invited authors to submit work in parallel with the SIAM Workshop on Parameter Space Dimension Reduction, held July 9–10, 2017, in Pittsburgh, PA.

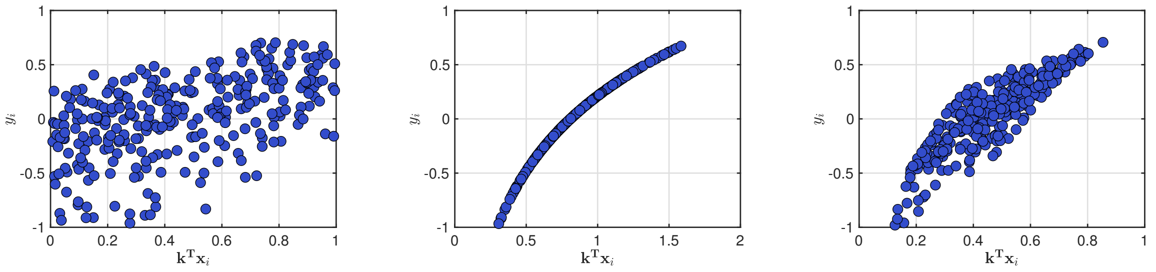
https://doi.org/10.1137/18M1168571

Funding: The first author's work was supported by a Rolls-Royce fellowship. The second author's work was supported by Magdalene College, Cambridge.

[†] Postdoctoral Fellow, Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK, and Group Leader, Data-Centric Engineering, The Alan Turing Institute, London, UK (ps583@cam.ac.uk, pseshadri@turing.ac.uk).

[‡] Undergraduate Research Assistant, Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK (sy334@cam.ac.uk).

[§] Reader, Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK (gtp10@cam.ac.uk).



(a) (b) (c)

Figure 1. *Sufficient summary plots of the function* $y = log(x_1 + x_2 + x_3)$*, where the horizontal axis is given by* $\mathbf{k}^T(x_1,\ x_2,\ x_3)$*. In* (a) $\mathbf{k} = (1, 0, 0)^T$*, in* (b) $\mathbf{k} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^T$*, while in* (c) $\mathbf{k} = (0.2, 0.5, 0.2)^T$.

dimension reduction range from airfoil aerodynamics [21] and combustion [27] to economic forecasting (see page 4403 in [2]), turbomachinery aerothermodynamics [34], solar energy [10], and epidemiology [15].

To motivate an exposition of the relevant literature and to set the stage for our paper, we introduce the trivariate function $y = \log(x_1+x_2+x_3)$. In Figure 1, we generate scatter plots of the function on three different subspaces $\mathbf{k}^T\mathbf{x}$—i.e., linear combinations of the three variables x_1, x_2, and x_3. In (a) $\mathbf{k}$ is one of the canonical vectors, in (b) all of the elements of $\mathbf{k}$ are equivalent (and normalized), while in (c) we moderately perturb the entries in (b). Naturally, we find the subspace $\mathbf{k}^T\mathbf{x}$ in (b) to be *optimal* in the sense that it most accurately characterizes the three-dimensional (3D) function as a function of one variable. Scatter plots of the kind in Figure 1, where the horizontal axis is a subspace of the inputs, are known as *sufficient summary plots* [13] or shadow plots. These plots are useful in identifying low-dimensional structure in high-dimensional problems. However, as the selection of $\mathbf{k}$ in Figure 1 illustrates, the impediment to realizing this identification is an appropriate choice of the subspace.

Enter *active subspaces* [7], a collection of ideas that facilitate subspace-based dimension reduction for deterministic computer models. If one can compute gradients of a function f, or even approximate them, then one can identify directions along which (on average) f exhibits the greatest variation, and conversely directions along which (on average) f is near constant.[1] Active subspaces owe their development to initial work by Samarov [33] and more recent work by Constantine et al. (see [6, 8, 7]). The latter references build the theory of active subspaces and also offer practical algorithms for their computation. Parallels between active subspaces and *ridge functions* have also been recently explored in the literature [9].

The topic of ridge functions [28] may be interpreted as a broad generalization of ideas within active subspaces. They are of the form $f(\mathbf{x}) = g(\boldsymbol{M}^T\mathbf{x})$, where $\boldsymbol{M}$ is a tall matrix, implying that g is a function of fewer variables than f. Functions of this form are known as *generalized ridge functions*. When $d = 1$, Pinkus [28] defines these functions as *ridge functions*. The elements of $\boldsymbol{M}$ and the function g are useful in identifying low-dimensional structures, in a similar vein to the example discussed above. In the case where $\boldsymbol{M}$ is a vector, which

[1] Assuming, of course, f admits an active subspace; there is no guarantee that every differentiable multivariate function f admits an active subspace.

implies that g is univariate function, Cohen et al. [5] provide estimates on how well f can be approximated using only point evaluations. Building on their work, Fornasier, Schnass, and Vybiral [17] study the case where $\boldsymbol{M}$ is a matrix and provide two randomized algorithms for approximating $g(\boldsymbol{M}^T\mathbf{x})$. The work of Tyagi and Cevher [36] is also similar in scope; they provide an algorithm that approximates functions of the form $f(\mathbf{x}) = \sum_{i=1}^{n} g_i(\boldsymbol{M}_i^T\mathbf{x})$. Clearly, there are similarities between ridge functions, projection pursuit regression (see page 390 of [22]), and, by association, single layer neural networks. The projection pursuit regression problem can be interpreted as an approximation with

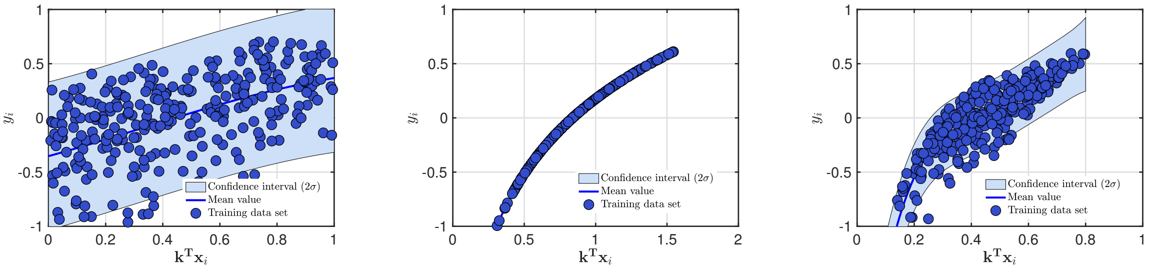
$$\sum_{i=1}^{k} g_i \left(\mathbf{m}_i^T \mathbf{x}\right), \tag{1}$$

where $\mathbf{m}_i \in \mathbb{R}^d$ and the univariate function g_i are unknown, determined via a stepwise greedy algorithm following Friedman and Stuetzle [18]. It is worth emphasizing that although our function may not admit an analytical ridge structure, i.e., $f(\mathbf{x}) \neq g(\boldsymbol{M}^T\mathbf{x})$, one may still be able to approximate $f(\mathbf{x}) \approx g(\boldsymbol{M}^T\mathbf{x})$ extracting useful inference. We refer to this as a *ridge approximation*, which is obtained by evaluations of f and by minimizing a suitably constructed objective function to ascertain $\boldsymbol{M}$ and g.

In introducing methods for approximation and statistical regression (projection pursuit regression), it is important to note the differences between both paradigms. In statistical regression, as alluded to previously, $f(\mathbf{x})$ is not deterministic and therefore has a conditional density that is not a delta function [7]. Furthermore, the joint density of $\mathbf{x}$ is not known in the regression setting. In comparison, when working with computer models—i.e., in the approximation paradigm—the joint density of $\mathbf{x}$ is known and often prescribed according to some optimal design of experiment, for example, Latin hypercube sampling, D-optimal design, etc. Last, techniques for estimating gradients for unknown regression functions are notoriously expensive in high dimensions. Thus, non–gradient-based approaches are sought.

The field of sufficient dimension reduction is rich in ideas for achieving subspace-based dimension reduction within a statistical regression framework. The goal here is to replace the vector of covariates (inputs) with their projection onto a subspace assembled from the space of the covariates themselves [14]. This subspace must be constructed without loss of information based either on the conditional mean, conditional variance, or, more generally, the conditional distribution of the outputs with respect to the inputs. The numerous methods for estimating these subspaces include sliced inverse regression (SIR) [24], sliced average variance estimation (SAVE) [11], and contour regression (CR) [23], to name but a few (see [26] for a detailed review).

We now return to the sufficient summary plots in Figure 1. Assume we did not know the true function $f(x) = \log(x_1 + x_2 + x_3)$ and that subfigures (a), (b), and (c) were simply the outcome of any one of the aforementioned algorithms. How do we determine that the subspace in (b) is ideally suited for the data? The logic we pursue in the paper is as follows. Assume we utilize Gaussian process regression (see Chapter 2 of [32]) to estimate the mean and standard deviation of the response y_i, as shown in Figure 2. Then our criterion for selecting $\mathbf{k}$ in (b) is clear: it reduces the Gaussian process yielded standard deviation. In other words, for the same value of $\mathbf{x}_i$ there is only one unique y_i, ensuring that the deviation from the mean is



(a) (b) (c)

Figure 2. *Sufficient summary plots of the function* $y = log(x_1 + x_2 + x_3)$*, where the horizontal axis is given by* $\mathbf{k}^T(x_1,\ x_2,\ x_3)$ *with a Gaussian process regression response surface. In* (a) $\mathbf{k} = (1, 0, 0)^T$*, in* (b) $\mathbf{k} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^T$*, while in* (c) $\mathbf{k} = (0.2, 0.5, 0.2)^T$*.*

near zero.

There are two key ideas that emerge from this discussion. The first is that the posterior mean of a Gaussian process computed on a dimension-reducing subspace is a good candidate for g. In fact, this idea has been previously studied in [35] and [25], albeit using different techniques than those presented here. The second idea is that the suitability of a dimension-reducing subspace is reflected by the posterior variance: should the posterior variance be too large, as in Figure 2(a), then one may need to opt for a different choice of $\mathbf{k}$.

In a nutshell, our objective in this paper is to offer an algorithm for computing a dimension-reducing subspace, under the constraint that g is the posterior mean of a Gaussian process on this subspace. The remainder of this paper is structured as follows. In section 2 we survey the connections between ridge, active, and sufficient dimension-reduction subspaces. Then, motivated by the techniques in [35] and [9], we introduce an algorithm for computing a ridge subspace that uses tools from Gaussian process regression and manifold optimization. This is followed by numerical examples in section 4.

2. Ridge functions.

If there exist a function $f : \mathbb{R}^d \to \mathbb{R}$ and a matrix $\boldsymbol{M} \in \mathbb{R}^{d\times m}$, with $m \leq d$, that satisfy

$$f(\mathbf{x}) = g(\boldsymbol{M}^T\mathbf{x}) \tag{2}$$

for a function $g : \mathbb{R}^m \to \mathbb{R}$, then f is a *ridge function* [28]. We call the subspace associated with the span of $\boldsymbol{M}$ its *ridge subspace*, denoted by $\mathcal{S}(\boldsymbol{M})$. We also assume that the columns of $\boldsymbol{M}$ are orthonormal, i.e., $\boldsymbol{M}^T\boldsymbol{M} = \boldsymbol{I}$. The above definitions imply that the gradient of f is zero along directions that are orthogonal to $\mathcal{S}(\boldsymbol{M})$. In other words, if we replace $\mathbf{x}$ with $\mathbf{x} + \mathbf{h}$ where $\boldsymbol{M}^T\mathbf{h} = 0$, then it is trivial to see that $f(\mathbf{x} + \mathbf{h}) = g(\boldsymbol{M}^T(\mathbf{x} + \mathbf{h})) = f(\mathbf{x})$.

2.1. Computing the optimal subspace.

So, given point evaluations of f, how does one compute $\boldsymbol{M}$? Also, what are the properties of $\boldsymbol{M}$? In their paper, Constantine et al. [9] draw our attention to the conditional and marginal densities defined on the subspace coordinates of the ridge subspace and its orthogonal complement. They present two main ideas.

The first is the *orthogonal invariance* associated with $\boldsymbol{M}$, i.e., we are not interested in estimating a particular $\boldsymbol{M}$, but rather the subspace of $\boldsymbol{M}$—i.e., the ridge subspace. To study this further, we place a few additional assumptions on (2). Let the joint probability density function $\rho(\mathbf{x})$ and marginal densities ρ_i be related by

$$\boldsymbol{\rho}(\mathbf{x}) = \prod_{i=1}^{d} \rho(x^{(d)}). \tag{3}$$

We assume that f is square integrable with respect to this probability density function:

$$\int_{\boldsymbol{\rho}} f^2(\mathbf{x})\boldsymbol{\rho}(\mathbf{x})d\mathbf{x} < \infty, \tag{4}$$

where the domain of f is the support of $\boldsymbol{\rho}$. Define $\mathcal{S}(\boldsymbol{N})$ to be the orthogonal complement of the ridge subspace, where

$$\boldsymbol{N} = (\text{null}(\boldsymbol{M}^T)) \quad \text{with} \quad \boldsymbol{N} \in \mathbb{R}^{d\times(d-m)}. \tag{5}$$

Placing the columns of $\boldsymbol{M}$ and $\boldsymbol{N}$ together in a single matrix,

$$\boldsymbol{G} = \left[\begin{array}{cc} \boldsymbol{M} & \boldsymbol{N} \end{array}\right], \tag{6}$$

readily implies that $\boldsymbol{G}\boldsymbol{G}^T = \boldsymbol{I}$. We now write the full decomposition of f as

$$f(\mathbf{x}) = f\left(\boldsymbol{G}\boldsymbol{G}^T\mathbf{x}\right) = f\left(\boldsymbol{M}\boldsymbol{M}^T\mathbf{x} + \boldsymbol{N}\boldsymbol{N}^T\mathbf{x}\right) = f\left(\boldsymbol{M}\mathbf{u} + \boldsymbol{N}\mathbf{v}\right), \tag{7}$$

with the following subspace coordinates:

$$\mathbf{u} \in \mathbb{R}^m \quad \text{and} \quad \mathbf{v} \in \mathbb{R}^{d-m}, \tag{8}$$

where

$$\mathbf{u} = \boldsymbol{M}^T\mathbf{x} \quad \text{and} \quad \mathbf{v} = \boldsymbol{N}^T\mathbf{x}. \tag{9}$$

It should be noted here that if f admits a ridge function structure, then the gradient of f along the directions $\boldsymbol{N}\mathbf{v}$ are expected to be zero.

For a fixed subspace coordinate $\mathbf{u}$, Constantine et al. [9] write the conditional expectation of f to be

$$\mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right) = \int f\left(\boldsymbol{M}\mathbf{u} + \boldsymbol{N}\mathbf{v}\right)\pi\left(\mathbf{v}|\mathbf{u}\right)d\mathbf{v}, \tag{10}$$

where the conditional probability $\pi\left(\mathbf{v}|\mathbf{u}\right)$ can be expressed in terms of its marginal $\pi\left(\mathbf{u}\right)$ and $\pi\left(\mathbf{u}, \mathbf{v}\right)$ joint probabilities:

$$\pi\left(\mathbf{v}|\mathbf{u}\right) = \frac{\pi\left(\mathbf{u}, \mathbf{v}\right)}{\pi\left(\mathbf{u}\right)} = \frac{\rho\left(\boldsymbol{M}\mathbf{u} + \boldsymbol{N}\mathbf{v}\right)}{\int \rho\left(\boldsymbol{M}\mathbf{u} + \boldsymbol{N}\mathbf{v}\right)d\mathbf{v}}. \tag{11}$$

To demonstrate the orthogonal invariance associated with $\boldsymbol{M}$ we replace it with $\boldsymbol{MQ}$, where $\boldsymbol{Q} \in \mathbb{R}^{m \times m}$ is an orthogonal matrix. Plugging this into (10) yields

$$\mathbb{E}\left(f|\mathbf{u}, \boldsymbol{MQ}\right) = \int f\left(\boldsymbol{MQ}\left(\boldsymbol{MQ}\right)^T \mathbf{x} + \boldsymbol{N}\mathbf{v}\right) \pi\left(\mathbf{v}|\mathbf{u}\right) d\mathbf{v} \tag{12}$$

$$= \int f\left(\boldsymbol{M}\underbrace{\boldsymbol{QQ}^T}_{\boldsymbol{I}}\boldsymbol{M}^T\mathbf{x} + \boldsymbol{N}\mathbf{v}\right) \pi\left(\mathbf{v}|\mathbf{u}\right) d\mathbf{v} \tag{13}$$

$$= \int f\left(\boldsymbol{M}\mathbf{u} + \boldsymbol{N}\mathbf{v}\right) \pi\left(\mathbf{v}|\mathbf{u}\right) d\mathbf{v}. \tag{14}$$

It can be easily shown that the conditional probability also remains the same under the orthogonal transformation $\boldsymbol{MQ}$. This demonstrates that the particular choice of $\boldsymbol{M}$ does not matter, but rather its subspace does. In [9], the authors draw our attention to Theorem 8.3 in Pinkus [28], which proves that $\mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right)$ is the *optimal ridge profile*—a function—in the L_2 norm.

So, how do we compute this *optimal subspace* associated with this ridge profile? The second point made in [9] is that this subspace can be computed by solving the following quadratic form:

$$\begin{aligned} &\underset{\boldsymbol{M}}{\text{minimize}} \quad \omega(\boldsymbol{M}) = \frac{1}{2}\int_{\boldsymbol{\rho}} \left(f\left(\mathbf{x}\right) - \mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right)\right)^2 \boldsymbol{\rho}\left(\mathbf{x}\right) d\mathbf{x} \\ &\text{subject to} \quad \boldsymbol{M} \in \mathbb{G}\left(m, d\right). \end{aligned} \tag{15}$$

The constraint in (15) restricts $\boldsymbol{M}$ to be from the subspace $\mathbb{G}(m, d)$, which denotes the space of m-dimensional subspaces of $\mathbb{R}^d$, i.e., the *Grassman* manifold (see page 30 in [1]). Recognizing (15) as a manifold optimization problem permits us to compute the gradients on the Grassman manifold. Following page 321 in [16], and denoting the gradient on the Grassman manifold by the symbol $\nabla_{\mathbb{G}}$, we have

$$\nabla_{\mathbb{G}}\omega\left(\boldsymbol{M}\right) = \int_{\boldsymbol{\rho}} \left(f\left(\mathbf{x}\right) - \mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right)\right)\left(-\nabla_{\mathbb{G}}\mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right)\right) \boldsymbol{\rho}\left(\mathbf{x}\right) d\mathbf{x} \tag{16}$$

$$= \int_{\rho} \left(\mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right) - f\left(\mathbf{x}\right)\right)\left(\boldsymbol{I} - \boldsymbol{MM}^T\right) \frac{\partial}{\partial \boldsymbol{M}}\mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right) \boldsymbol{\rho}\left(\mathbf{x}\right) d\mathbf{x}. \tag{17}$$

A few remarks regarding this optimization problem are in order. First, to compute the derivative inside the integral, we require $f \in C^1\left(\mathbb{R}^d\right)$, which denotes a class of real-valued functions with continuous first derivatives. Provided the gradients $\frac{\partial}{\partial \boldsymbol{M}}\mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right) \in \mathbb{R}^{m \times d}$ can be determined, solutions to this optimization problem can be computed using an appropriate gradient-based optimizer. In subsection 3.3 we use tools from Gaussian processes to estimate these gradients. Second, (15) is not necessarily convex; therefore, we are not guaranteed a unique global minimum during an optimization. However, as we demonstrate in section 4, near local optimal solutions can be computed.

Another salient point concerns the choice of the manifold itself. In using the Grassman manifold, we enforce the assumption of *homogenity*, i.e., $\omega\left(\boldsymbol{M}\right) \equiv \omega\left(\boldsymbol{MQ}\right)$ implying—as stated previously—that the specific entries in $\boldsymbol{M}$ do not matter, but rather its subspace does. In other words, our objective is invariant to any choice of the basis [16]. In section 3 we discuss the ramifications of this assumption when using Gaussian process regression.

2.2. Connections between ridge subspaces and sufficient dimension reduction.

One approach to estimate the ridge subspace is to develop a computational method that implements (15). In subsection 3.2 we detail such a procedure using Gaussian process regression. However, other algorithms that utilize ideas from sufficient dimension reduction theory can also be leveraged. The key is to understand the relationship between the subspace computed from techniques like SIR, SAVE, and CR and the *ridge subspace*. As a prelude to exploring these connections, it is important to distinguish statistical regression from approximation.

In the regression setting we are given independent and identically distributed random variables $(\mathbf{x}_i, y_i)$ from an *unknown* joint distribution $\pi(\mathbf{x}, y)$, where $i = 1, \ldots, N$ for some N. Our objective is to characterize the conditional dependence of y on $\mathbf{x}$ either through its expectation $\mathbb{E}\left(y|\mathbf{x}\right)$ or its variance $\sigma^2\left(y|\mathbf{x}\right)$. This requires a model $y = h(\mathbf{x}) + \epsilon$ for predicting y given $\mathbf{x}$, where ϵ is a zero-mean random error that is independent of $\mathbf{x}$. This implies that for the same $\mathbf{x}_*$ we have multiple values of y_*. Here h is a parametric function and will typically be obtained by penalizing the errors in prediction via an appropriate *loss function*, the most common one being the l_2 norm error [22]. In the approximation setting, we are given input / output pairs $(\mathbf{x}_i, y_i)$ obtained by evaluating a computer model under a known density $\pi(\mathbf{x})$ via an appropriately chosen *design of experiment*. These inputs may be either design variables, boundary conditions, or state variables. As the computer model, in general, does not have any random error associated with it, multiple evaluations for a particular $\mathbf{x}_*$ will always yield the same y_*. In this context, our objective is to obtain a useful approximation $y = f(\mathbf{x})$ valid for *all* $\mathbf{x}$ over $\pi(\mathbf{x})$.

Now, despite these differences, ideas from sufficient dimension reduction can be used for approximation. Glaws, Constantine, and Cook [19] detail the mathematical nuances involved in doing so for SIR and SAVE. They prove that that the conditional independence of the inputs and outputs in the approximation setting—i.e., for a deterministic function—is equivalent to the function being a ridge function (see Theorem 6 in [19]). But is there any benefit in using SIR, SAVE, and other sufficient dimension reduction methods for dimension reduction when approximating? Adragni and Cook [2] suggest that moment-based sufficient dimension reduction techniques such as SIR and SAVE, which are designed to provide estimates of the minimum sufficient linear reduction, are not tailored for prediction. They further suggest that model-based inverse regression techniques, such as those presented in [12], may be more useful. In section 4 we test their suggestion by comparing the results of our proposed algorithm with SIR, SAVE, and CR.

2.3. Connections between ridge subspaces and active subspaces.

Active subspaces [7] refer to a set of ideas for parameter-based dimension reduction in the *approximation* setting (see subsection 2.2). To compute the active subspace of a function f, we require that f be differentiable and Lipschitz continuous—implying that the norm of its gradient $\nabla f_{\mathbf{x}}$ is bounded. Then one assembles the covariance matrix $\boldsymbol{C}$,

$$\boldsymbol{C} = \int \nabla f_{\mathbf{x}} \nabla f_{\mathbf{x}} \rho\left(\mathbf{x}\right) d\mathbf{x}, \tag{18}$$

also known as the outer product of the gradient matrix.[2] What follows is an eigenvalue decomposition of $\boldsymbol{C}$, where the first m eigenvectors are used to define the *active* subspaces and remaining $d - m$ eigenvectors the *inactive* subspaces; the theorem below connects active subspaces and ridge subspaces.

Theorem 1. *If we assume that $\boldsymbol{C}$ admits the eigendecomposition*

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{N} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{M} & \boldsymbol{N} \end{bmatrix}^T, \quad \textit{where} \quad \boldsymbol{\Lambda}_1 \in \mathbb{R}^{m \times m}, \tag{19}$$

then

$$\left(\int \left(f\left(\mathbf{x}\right) - \mathbb{E}\left(f | \boldsymbol{M}^T \mathbf{x}\right)\right)^2 \rho\left(\mathbf{x}\right) d\mathbf{x} \right)^{\frac{1}{2}} \leq C \ \left(\textit{trace}\left(\Lambda_2\right)\right)^{\frac{1}{2}}, \tag{20}$$

where C is a constant that depends on $\rho(\mathbf{x})$.

The proof follows from Theorem 4.4 of [7]. It is important to emphasize that, in practice, the right-hand side of (20) is difficult to compute because we do not have access to the eigenvalues associated with the covariance matrix that yields eigenvectors $[\boldsymbol{M}, \boldsymbol{N}]$. Furthermore, while the decay in the eigenvalues above can be used to set the size of m when using active subspaces, when approximating ridge subspaces, one has to opt for other criteria—for instance, the authors in [35] utilize a Bayesian information criterion to determine m.

3. Gaussian processes.

The definition of a Gaussian process is a "collection of random variables, any finite number of which have a joint Gaussian distribution" [32]. It is completely defined by its mean and covariance functions. Restating the definition in the form we need, we have

$$f\left(\mathbf{x}\right) \sim \mathcal{GP}\left(\mu\left(\mathbf{u}\right), \boldsymbol{k}\left(\mathbf{u}\right)\right), \tag{21}$$

where μ is the mean function and $\boldsymbol{k}$ represents a parameterized covariance matrix, albeit with a slight abuse of notation. It is important to note that we define our Gaussian process on the reduced coordinates $\mathbf{u} = \boldsymbol{M}^T \mathbf{x}$.

In this section we have two main goals: first, to define a Gaussian ridge function and to outline its properties; second, to provide an algorithm that computes a Gaussian ridge function. To ease our exposition, we assume the existence of the following sets:

- A set of input/output training data pairs:

$$\left(\hat{\mathbf{x}}_i, \hat{f}_i\right) \quad \text{for } i = 1, \ldots, N_{train},$$

$$\hat{\mathbf{f}} = \begin{pmatrix} \hat{f}_1 \\ \vdots \\ \hat{f}_{N_{train}} \end{pmatrix}, \quad \hat{\boldsymbol{U}} = \begin{pmatrix} \hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_N \end{pmatrix}^T, \quad \text{and} \quad \hat{\mathbf{u}}_i = \boldsymbol{M}^T \hat{\mathbf{x}}_i. \tag{22}$$

[2] Note that this is a symmetric positive semidefinite matrix.

- A set of input/output testing data pairs:

$$\left(\tilde{\mathbf{x}}_i, \tilde{f}_i\right) \quad \text{for } i = 1, \ldots, N_{test},$$

$$\tilde{\mathbf{f}} = \begin{pmatrix} \tilde{f}_1 \\ \vdots \\ \tilde{f}_{N_{test}} \end{pmatrix}, \quad \tilde{\boldsymbol{U}} = \left(\begin{array}{ccc} \tilde{\mathbf{u}}_1, & \ldots, & \tilde{\mathbf{u}}_N \end{array} \right)^T, \quad \text{and} \quad \tilde{\mathbf{u}}_i = \boldsymbol{M}^T \tilde{\mathbf{x}}_i. \tag{23}$$

3.1. Gaussian ridge functions (posterior mean).

Succinctly stated, a Gaussian ridge function is the posterior mean of the Gaussian process in (21), written as

$$\bar{g}\left(\mathbf{u}\right) = \boldsymbol{k}\left(\mathbf{u}, \hat{\boldsymbol{U}}\right)\boldsymbol{\beta}. \tag{24}$$

It is a linear sum of kernel functions with coefficients $\boldsymbol{\beta}$—an outcome of the representer theorem (see page 132 of [32]). Before we define these coefficients, we first focus on the covariance term on the right-hand side in (24). In what follows, we employ the squared exponential covariance kernel

$$k\left(\mathbf{u}_i, \mathbf{u}_j\right) = \sigma_f^2 \exp\left(-\frac{1}{2}\left(\mathbf{u}_i - \mathbf{u}_j\right)^T \boldsymbol{\Gamma}^{-1}\left(\mathbf{u}_i - \mathbf{u}_j\right)\right), \tag{25}$$

where the diagonal matrix $\boldsymbol{\Gamma}$ is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} l_1^2 & & 0 \\ & \ddots & \\ 0 & & l_m^2 \end{bmatrix}. \tag{26}$$

The constant σ_f^2 is known as the *signal variance* and constants $l_1, \ldots, l_m$ are the *correlation lengths* along each of the m coordinates of $\mathbf{u}$. The parameterized covariance matrix $\boldsymbol{k}\left(\mathbf{u}, \hat{\boldsymbol{U}}\right) \in \mathbb{R}^{1\times N}$ in (24) can now be written as

$$\boldsymbol{k}\left(\mathbf{u}, \hat{\boldsymbol{U}}\right) = \left(\begin{array}{ccc} k\left(\mathbf{u}, \hat{\mathbf{u}}_1\right), & \ldots, & k\left(\mathbf{u}, \hat{\mathbf{u}}_N\right) \end{array} \right). \tag{27}$$

The vector of coefficients $\boldsymbol{\beta} \in \mathbb{R}^N$ is given by

$$\boldsymbol{\beta} = \left(\boldsymbol{K} + \sigma_n \boldsymbol{I}\right)^{-1} \hat{\mathbf{f}}, \tag{28}$$

where the (i, j)th entry of the matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ is given by

$$\boldsymbol{K}(i, j) = k\left(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j\right); \tag{29}$$

in other words it depends only on the training data set. Once again, for notational convenience, we define the posterior mean values at the training and testing points by

$$\bar{\mathbf{g}}_{train} = \begin{pmatrix} \bar{g}\left(\hat{\mathbf{u}}_1\right) \\ \vdots \\ \bar{g}\left(\hat{\mathbf{u}}_{N_{train}}\right) \end{pmatrix}, \quad \bar{\mathbf{g}}_{test} = \begin{pmatrix} \bar{g}\left(\tilde{\mathbf{u}}_1\right) \\ \vdots \\ \bar{g}\left(\tilde{\mathbf{u}}_{N_{test}}\right) \end{pmatrix}. \tag{30}$$

Typically, the cost of constructing a Gaussian process is dominated by an $\mathcal{O}\left(N_{train}^3\right)$ linear solve operation. By defining our Gaussian process over a reduced set of coordinates, there are two instances where we reduce the computational cost:

1. We reduce the number of coordinates where our kernel has to be evaluated, which happens $\mathcal{O}\left(N_{train}^2\right)$ times.
2. We reduce the number of hyperparameters that need to be optimized.

The first instance is dominated by the linear solve, while the second is harder to analyze and will depend on whether there is a hyparameter associated with each dimension d.

To aid the optimization strategy we describe in the forthcoming section, we are interested in obtaining the gradients of (24); this yields

$$\begin{aligned} \frac{\partial \bar{g}}{\partial \mathbf{u}} &= \frac{\partial \boldsymbol{k}\left(\mathbf{u}, \hat{\boldsymbol{U}}\right)}{\partial \mathbf{u}} \boldsymbol{\beta} \\ &= -\boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{U}} \left(\boldsymbol{k}\left(\mathbf{u}, \hat{\boldsymbol{U}}\right)^T \odot \boldsymbol{\beta} \right), \end{aligned} \tag{31}$$

where the symbol $\odot$ indicates an elementwise product. In practice, any differentiable kernel may be used to compute the gradient of the Gaussian ridge function. Now, in using a Gaussian ridge function, we make the assertion that it is approximately the expectation of f projected on $\mathbf{u} = \boldsymbol{M}^T \mathbf{x}$, i.e.,

$$\mathbb{E}\left(f | \mathbf{u}, \boldsymbol{M}\right) \approx \bar{g}\left(\mathbf{u}\right). \tag{32}$$

Its gradients can then be computed via

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{M}} \bar{g}\left(\mathbf{u}\right) &= \frac{\partial \bar{g}}{\partial \mathbf{u}} \frac{\partial \left(\boldsymbol{M}^T \mathbf{x}\right)}{\partial \boldsymbol{M}} \\ &= -\boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{U}} \left(\boldsymbol{k}\left(\mathbf{u}, \hat{\boldsymbol{U}}\right)^T \odot \boldsymbol{\beta} \right) \mathbf{x}. \end{aligned} \tag{33}$$

It is clear that the quality of our approximation in (32) is contingent upon a suitable choice of the hyperparameters

$$\boldsymbol{\theta} = \left\{ \sigma_f^2, \sigma_n^2, l_1, \ldots, l_m \right\}. \tag{34}$$

A well-worn heuristic to estimate these hyperparameters, which in itself is a nonconvex optimization problem, is to maximize the marginal likelihood (see page 113 of [32])

$$\begin{aligned} &\underset{\boldsymbol{\theta}}{\text{maximize}} && \log p\left(\mathbf{u}, \boldsymbol{\theta}\right) \\ &\text{subject to} && \log p\left(\mathbf{u}, \boldsymbol{\theta}\right) = -\frac{1}{2} \mathbf{f}^T \left(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}\right)^{-1} \mathbf{f} - \frac{1}{2} \log \left| \boldsymbol{K} + \sigma_n^2 \boldsymbol{I} \right| - \frac{N}{2} \log\left(2\pi\right). \end{aligned} \tag{35}$$

The first term in $\log\, p\left(\mathbf{u}, \boldsymbol{\theta}\right)$, which depends solely on the covariance matrix and values of f, is the *data-fit* term. The second term, which depends only upon the covariance function, is known as the *complexity penalty*, while the third term is simply a normalizing constant. The problem of computing the maximum in (35) can be readily solved via a gradient-based optimizer; formulas for the gradients of $\log\, p$ with respect to the hyperparameters are given in section 5.9 of [32].

3.2. The utility of the posterior variance.

In our presentation of Gaussian ridge functions so far we have not discussed the posterior variance. It is given by

$$\mathbb{V}\left[\mathcal{GP}\right] = k\left(\mathbf{u},\mathbf{u}\right) - k\left(\mathbf{u},\hat{\boldsymbol{U}}\right)^{T}\left(\boldsymbol{K}+\sigma_{n}^{2}\boldsymbol{I}\right)^{-1}\mathbf{k}\left(\mathbf{u},\hat{\boldsymbol{U}}\right), \tag{36}$$

and can be very useful in quantifying the suitability of a particular choice of $\boldsymbol{M}$. Recall the plots in Figures 1 and 2. Our perception of a successful dimension reduction strategy through a sufficient summary plot is intimately tied to the variance in f for a fixed coordinate in the dimension-reducing subspace. Here, we argue that the expectation of the posterior variance, $\mathbb{E}\left[\mathbb{V}\left(g\right)\right]$, can be used to gauge the overall effectiveness of the subspace approximant $\hat{\boldsymbol{M}}$.

We make the above argument for two reasons. First, different dimension reduction methods—tailored for the same projection $\mathbb{R}^{d} \rightarrow \mathbb{R}^{m}$—will likely yield different subspaces. In [14] a clear distinction is made between dimension reduction techniques that identify a *central subspace* from those that find a *central mean subspace*; an algorithm for computing the *central variance subspace* is introduced in [37]. Our hypothesis is that regardless of the subspace sought, to infer the suitability of a particular $\hat{\boldsymbol{M}}$ as a ridge subspace we must be able to compare different methods that by definition will yield different subspaces. To this end, while a metric based on the subspace angle ϕ (see page 329 of [20])—i.e.,

$$\begin{aligned} \text{dist}\left(\boldsymbol{M},\tilde{\boldsymbol{M}}\right) &= \left\|\boldsymbol{M}\boldsymbol{M}^{T}-\tilde{\boldsymbol{M}}\tilde{\boldsymbol{M}}^{T}\right\|_{2} \\ &= \sin(\phi) \end{aligned} \tag{37}$$

—is useful for comparing the *distance* between the *ridge subspace* $\boldsymbol{M}$ and its approximant $\tilde{\boldsymbol{M}}$, it is limited to comparisons with one technique/subspace. The second point we articulate concerns cases when $m \geq 3$, where it is difficult to visually perceive how suitable a choice of $\bar{g}$ is. Here, the measure $\mathbb{E}\left[\mathbb{V}\left(g\right)\right]$ can be used (i) to quantify the error in the approximation, and (ii) to guide further computer simulations.

3.3. Computing the Gaussian ridge via manifold optimization.

Recall the definition of the correlation lengths associated with the Gaussian process regression model in (26). If we were to set the correlation lengths to be the same, i.e., $l_1 = l_2 = \cdots = l_m$, then we satisfy our requirement of homogenity and can optimize (15) using the gradient computation in (17). While this may be a potential solution strategy it does require us to modify the maximum likelihood criterion in (35). To circumvent this, we propose altering the constraint in (15); instead of optimizing over the Grassman manifold, we optimize over the Stiefel manifold,

$$\begin{aligned} \underset{\boldsymbol{M}}{\text{minimize}} \quad & \omega(\boldsymbol{M}) = \frac{1}{2}\int_{\boldsymbol{\rho}}\left(f\left(\mathbf{x}\right)-\bar{g}\left(\boldsymbol{M}^{T}\mathbf{x}\right)\right)^{2}\boldsymbol{\rho}\left(\mathbf{x}\right)d\mathbf{x} \\ \text{subject to} \quad & \boldsymbol{M}\in\mathbb{S}t\left(m,d\right). \end{aligned} \tag{38}$$

Here $\mathbb{S}t\left(m,d\right)$ is the set of all $d \times m$ orthogonal matrices

$$\mathbb{S}t := \left\{\boldsymbol{M}\in\mathbb{R}^{d\times m} : \boldsymbol{M}^{T}\boldsymbol{M} = \boldsymbol{I}_{m}\right\}; \tag{39}$$

it is an embedded submanifold of $\mathbb{R}^{d\times m}$ and is compact [1]. Denoting the gradients on the Stiefel manifold by $\tilde{\nabla}$, gradients of ω can be computed via

$$\nabla_{\mathbb{S}t}\omega\left(\boldsymbol{M}\right) = \int_{\boldsymbol{\rho}} \left(f\left(\mathbf{x}\right) - \bar{g}\left(\boldsymbol{M}^{T}\mathbf{x}\right)\right)\left(-\bar{\nabla}_{\mathbb{S}t}\bar{g}\left(\boldsymbol{M}^{T}\mathbf{x}\right)\right)\boldsymbol{\rho}\left(\mathbf{x}\right)d\mathbf{x} \tag{40}$$

$$= \int_{\rho} \left(\bar{g}\left(\boldsymbol{M}^{T}\mathbf{x}\right) - f\left(\mathbf{x}\right)\right)\left(\frac{\partial}{\partial \boldsymbol{M}}\bar{g}\left(\boldsymbol{M}^{T}\mathbf{x}\right) - \boldsymbol{M}\frac{\partial}{\partial \boldsymbol{M}}\bar{g}\left(\boldsymbol{M}^{T}\mathbf{x}\right)^{T}\boldsymbol{M}\right)\boldsymbol{\rho}\left(\mathbf{x}\right)d\mathbf{x} \tag{41}$$

(see section 2.53 of [16]). We use this expression in our algorithm for estimating $\boldsymbol{M}$ using a Gaussian ridge function. Ours is an iterative approach—optimizing over the Stiefel manifold for estimating $\boldsymbol{M}$ and then over the space of the hyperparameters for determining $\boldsymbol{\theta}$, where the correlation lengths along each of the m directions can be different. We remark here that our approach is similar to the alternating approach from Algorithm 2 in Constantine et al. [9] that uses polynomial approximations.

Our algorithmic workflow is given as follows. First, the data set must be split into training and testing data sets, where the former has N_{train} input/output pairs while the latter has N_{test} pairs. It is difficult to provide heuristics on what values to set for N_{test} and N_{train}; however, we do require that both testing and training sets are distributed with respect to the measure $\rho(\mathbf{x})$.

For our algorithm, we require an initial choice of the ridge $\boldsymbol{M}$, which is readily computed by applying a QR factorization on a random normal matrix and then taking the first m columns of $\boldsymbol{Q}$ (see lines 3–5 of Algorithm 1). This trick ensures that the columns of $\boldsymbol{M}$ are orthogonal. It also yields samples from the Stiefel manifold with uniform probability. As we do not know which local minima this algorithm will converge to, by sampling with uniform probability we avoid the possibility that the initialization will bias the algorithm towards a particular minimum. It also implies that each call of our algorithm will yield a different result, by virtue of the fact that the initial starting point—i.e., $\boldsymbol{M}_0$—is different. Following this, it is easy to project the d-dimensional data $\mathbf{x}$ to the m-dimensional space (line 7 of Algorithm 1).

Next, we train a Gaussian process model using $\left(\hat{\mathbf{u}}_i, \hat{\mathbf{f}}_i\right)$ and solve the optimization problem in (35) (line 7 of Algorithm 1) to obtain the hyperparameters $\boldsymbol{\theta}$. These values are then fed back into the Gaussian process posterior mean—i.e., our approximation of $\mathbb{E}\left(f|\mathbf{u}, \boldsymbol{M}\right)$—for prediction. In line 8 of Algorithm 1, we optimize over the Stiefel manifold using the hyperparameters $\boldsymbol{\theta}$. This requires discretizing the integral in (38), resulting in

$$\begin{aligned} r &= \frac{1}{2N_{test}}\sum_{i=1}^{N_{test}}\left(\tilde{f}_i - \bar{g}\left(\boldsymbol{M}^{T}\tilde{\mathbf{x}}_i, \boldsymbol{\theta}_*\right)\right)^2 \\ &= \frac{1}{2N_{test}}\left\|\tilde{\mathbf{f}} - \tilde{\mathbf{g}}\right\|_2^2. \end{aligned} \tag{42}$$

The gradients of this function on the Stiefel manifold can then be obtained using

$$\nabla_{\mathbb{S}t}r\left(\boldsymbol{M}\right) = \frac{1}{N_{test}}\sum_{i=1}^{N_{test}}\left(\tilde{f}_i - \bar{g}\left(\tilde{\mathbf{u}}_i, \theta\right)\right)\left\{\frac{\partial \bar{g}\left(\tilde{\mathbf{u}}_i\right)}{\partial \boldsymbol{M}} - \boldsymbol{M}\left(\frac{\partial \bar{g}\left(\tilde{\mathbf{u}}_i\right)}{\partial \boldsymbol{M}}\right)^{T}\boldsymbol{M}\right\}, \tag{43}$$

where $\partial\bar{g}\left(\tilde{\mathbf{u}}_i\right)/\partial\boldsymbol{M}$ is determined by plugging the testing data into (33). Both the values of r and $\nabla_{\mathbb{S}t}r \in \mathbb{R}^{d\times m}$ are provided to the manifold optimizer in step 8 in Algorithm 1. This

Algorithm 1. Gaussian ridge function approximation.

```
Data: Input/output pairs $(\mathbf{x}_j, f_j)$ where $j = 1, \ldots, N_{train} + N_{test}$
Result: Matrix $\boldsymbol{M}$
Split the data into training data set $(\hat{\mathbf{x}}_i, \hat{f}_i)$ where $i = 1, \ldots, N_{train}$.
Assemble the testing data set $(\tilde{\mathbf{x}}_i, \tilde{f}_i)$ where $i = 1, \ldots, N_{test}$.
Initialize a random normal matrix $\boldsymbol{A} \in \mathbb{R}^{d \times m}$.
Compute its QR factorization $\boldsymbol{A} = \left( \begin{array}{cc} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 \end{array} \right) \boldsymbol{R}$, where $\boldsymbol{Q}_1 \in \mathbb{R}^{d \times m}$ and $\boldsymbol{M} := \boldsymbol{Q}_1$.
while $|\triangle r| \leq \epsilon$ do
    Compute
        $$\hat{\mathbf{u}}_i = \boldsymbol{M}^T \hat{\mathbf{x}}_i \quad \text{and} \quad \tilde{\mathbf{u}}_i = \boldsymbol{M}^T \tilde{\mathbf{x}}_i.$$
    Solve the optimization problem using the training data $\left(\hat{\mathbf{u}}_i, \hat{\mathbf{f}}_i\right)$:
        $$\boldsymbol{\theta}_* = \text{argmax} \log p\left(\hat{\mathbf{u}}, \boldsymbol{\theta}\right).$$
    Solve the Stiefel manifold optimization problem using $\boldsymbol{\theta}_*$ on the testing data:
        $$\boldsymbol{M}^* = \text{argmin} \frac{1}{2N_{test}} \left\| \tilde{\mathbf{f}} - \tilde{\mathbf{g}}\left(\boldsymbol{M}, \boldsymbol{\theta}_*\right) \right\|_2^2.$$
    Set: $\boldsymbol{M} = \boldsymbol{M}^*$.
end
```

process is then repeated until the difference in residuals $|\triangle r|$ between successive iterations is below a chosen value ϵ.

4. Numerical studies.

Our codes for Algorithm 1 can be found at https://www.github.com/~psesh/Gaussian-Ridges. The codes are in `MATLAB` and require the `gpml` [31] toolbox for Gaussian processes and `manopt` [4], a set of utilities for manifold optimization. For the manifold optimization subroutine in our codes, we use the preconditioned Riemannian conjugate gradients algorithm of Boumal and Absil (see page 224 of [3]). For optimizing the hyperparameters in our Gaussian process models, we use the in-built (within `gpml`) conjugate gradient approach based on the Polak–Ribière method (see [29]). Initial values for all our hyperparameters were set to 0.

4.1. An analytical linear ridge.

In this section, we study the outcome of the algorithm provided in subsection 3.3 for discovering the ridge subspace associated with the simple problem

$$
\begin{aligned}
f(\mathbf{x}) = g\left(\boldsymbol{M}^T \mathbf{x}\right), \quad \text{where} \quad \boldsymbol{M} = \left[\begin{array}{cc} \mathbf{m}_1 & \mathbf{m}_2 \end{array}\right], \\
\boldsymbol{M} \in \mathbb{R}^{d \times 2}, \quad \mathbf{x} \in \mathbb{R}^d, \quad \text{and} \quad g : \mathbb{R}^2 \rightarrow \mathbb{R},
\end{aligned}
\tag{44}
$$

with $\mathbf{x} \in [-1, 1]^d$. Here g is a bivariate function given by

$$
g\left(y_1, y_2\right) = y_1 + y_2. \tag{45}
$$

We explore the result of the conjugate gradient optimizer by varying the number of training and testing points along with the number of dimensions d. As our optimization problem is nonconvex, we expect different outcomes based on the initial value provided to the optimizer. As a result, we run each optimization numerous times—using the same training and testing data set. In other words, each optimization trial begins with a random orthogonal matrix (see step 3 in Algorithm 1) and then aims to minimize our objective function (see step 8 in Algorithm 1). In the graphs below, we denote our optimized solution as

$$\boldsymbol{M}^{*} = \left[\begin{array}{cc} \mathbf{m}_1^{*} & \mathbf{m}_2^{*} \end{array} \right], \quad \text{where} \quad \boldsymbol{M}^{*} \in \mathbb{R}^{d \times 2}. \tag{46}$$

4.1.1. Case $d = 10$.

To begin, consider the isolated case where we set $N_{test} = 50$ and $N_{train} = 50$. Figure 3(a) illustrates the sufficient summary plot on $\boldsymbol{M}$, while in (b) we plot the sufficient summary plot on the optimizer-yielded $\boldsymbol{M}^{*}$, taken from a trial that yielded the lowest value of the objective function; clearly our algorithm offers a suitable approximation to this ridge function. We repeat this experiment 20 times, each time specifying a different initial orthogonal matrix. In Figure 3(c) we plot the change in the objective function over the number of iterations, and in (d) we plot the norm of the gradient per iteration.

As alluded to previously, it is difficult to offer bounds on the number of testing and training samples one should use in Algorithm 1. In Figure 3(d) we plot the probability of success for different combinations of testing and training samples for this problem. We declare an optimization trial to be successful if the objective function attains a value ≤ 0.005. Furthermore, as the outcome of any optimization is dependent upon the initial point, we repeat each experiment with 20 trials and report the average values in the individual circles in Figure 4(a). Circles that are black have, on average, a high chance of recovering the ridge subspace. For this example problem, it is clear that beyond $N_{test} \geq 50$ and $N_{train} \geq 50$, ridge recovery is obtained.

4.1.2. Case $d = 100$.

A similar plot for the case where $d = 100$ is shown in Figure 4(b). Here we observe that on average, beyond $N_{train} = 200$ we observe high recovery probabilities. Thus, we repeat the same set of experiments with $N_{test} = 300$ and $N_{train} = 300$ and plot the results in Figure 5. Once again our algorithm is able to identify the ridge structure associated with this problem. It should be noted that even with 300 testing and training samples, we do encounter solutions that do not converge, as shown in Figure 5(c) and (d). Our strategy for finding a suitable solution is to repeat a single experiment—fixing the number of training and testing samples—20 times and then select the solution that yields the lowest value of the objective function. This is one of the key shortcomings of our approach, i.e., the need to run several random trials and then select a solution. Selecting one of the optimization trials with poor convergence results in sufficient summary plots of the form in Figure 5(e). Here it is clear that the ridge structure is not identified.

4.2. A bivariate normal distribution.

In this example, we seek to approximate the function

$$f\left(\boldsymbol{M}^{T}\mathbf{x}\right) = \frac{1}{\sqrt{\left|\boldsymbol{\Sigma}\right|\left(2\pi\right)^{2}}} exp\left(-\frac{1}{2}\mathbf{u}\boldsymbol{\Sigma}^{-1}\mathbf{u}^{T}\right), \tag{47}$$

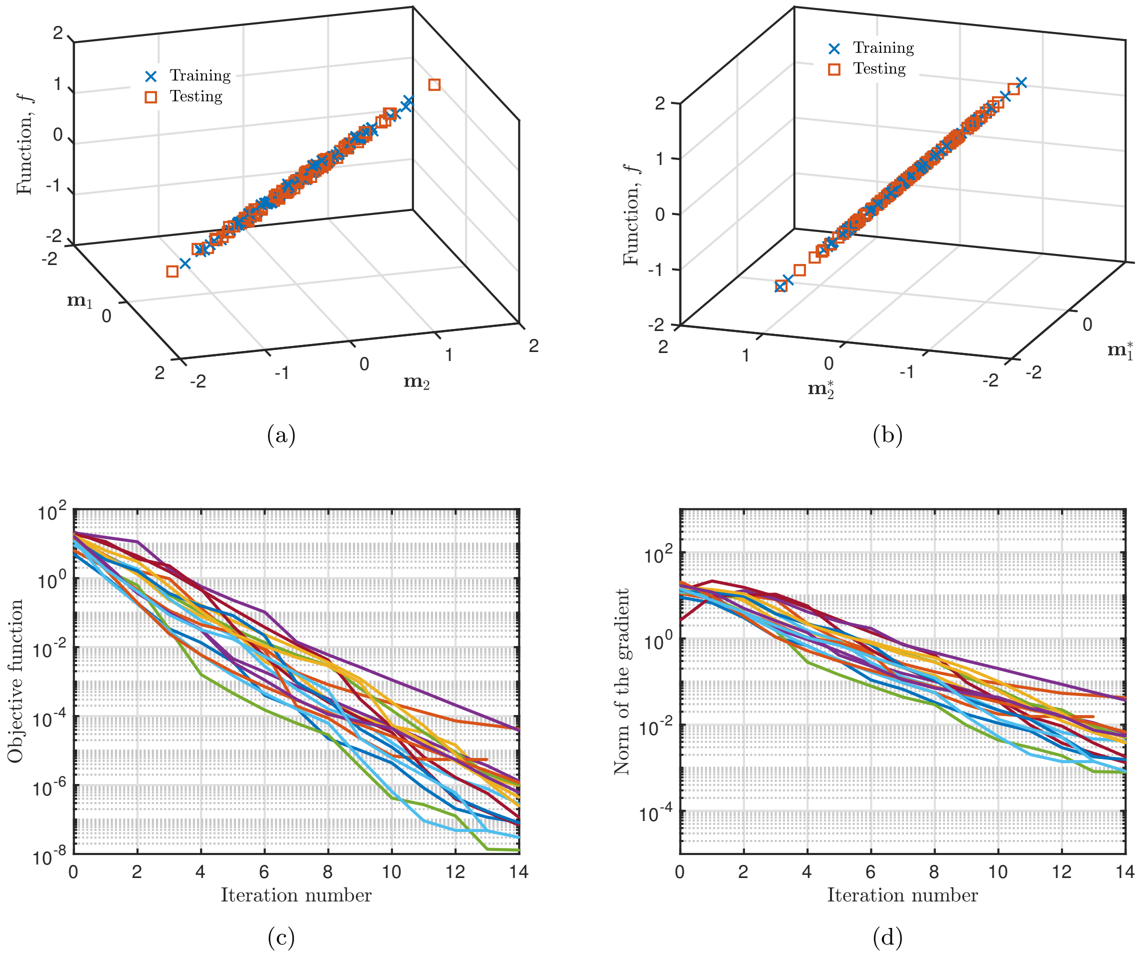


Figure 3. *Results of the analytical linear ridge problem with* $d = 10$*,* $N_{test} = 100$*, and* $N_{train} = 100$ *with* 20 *repetitions:* (a) *Sufficient summary plot on* $\boldsymbol{M}$*;* (b) *sufficient summary plot on* $\boldsymbol{M}^*$*;* (c) *objective function versus iterations;* and (d) *gradient norm versus iterations.*

where

$$\boldsymbol{M} = \begin{bmatrix} -0.5425 & 0.0654 \\ -0.6784 & -0.4931 \\ -0.2381 & -0.2367 \\ -0.1655 & 0.4643 \\ -0.4017 & 0.6935 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.25 & 0.30 \\ 0.30 & 1.0 \end{bmatrix}, \tag{48}$$

where identically distributed independent samples are taken from a multivariate normal distribution with zero mean and a variance of unity, i.e., $\mathbf{x} \in \mathcal{N}(0,1)^5$. Thus, our five-dimensional function is actually a multivariate Gaussian distribution over a two-dimensional subspace. In Figure 6 we compare the original function in (e) with two different approximations, shown in (a), (b) and (c), (d). These approximations use a different number of training and testing

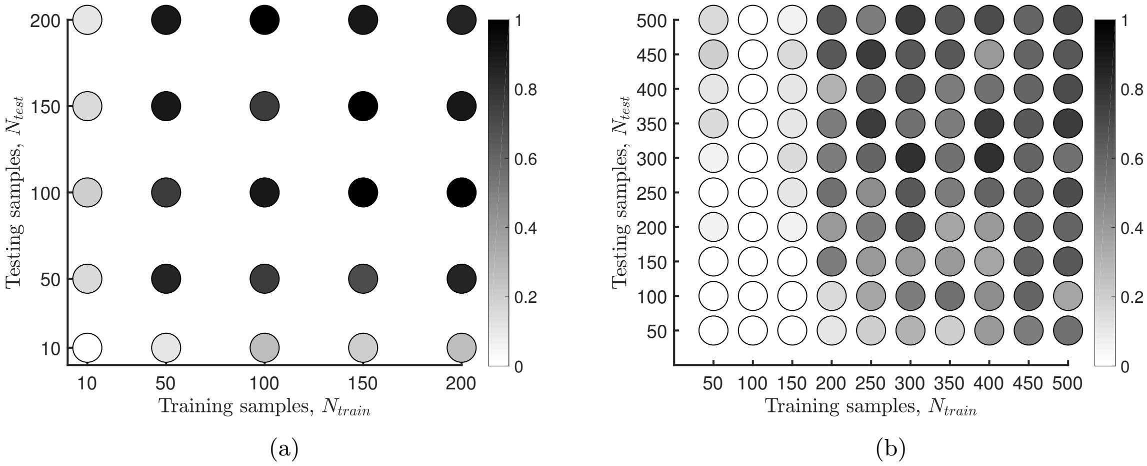


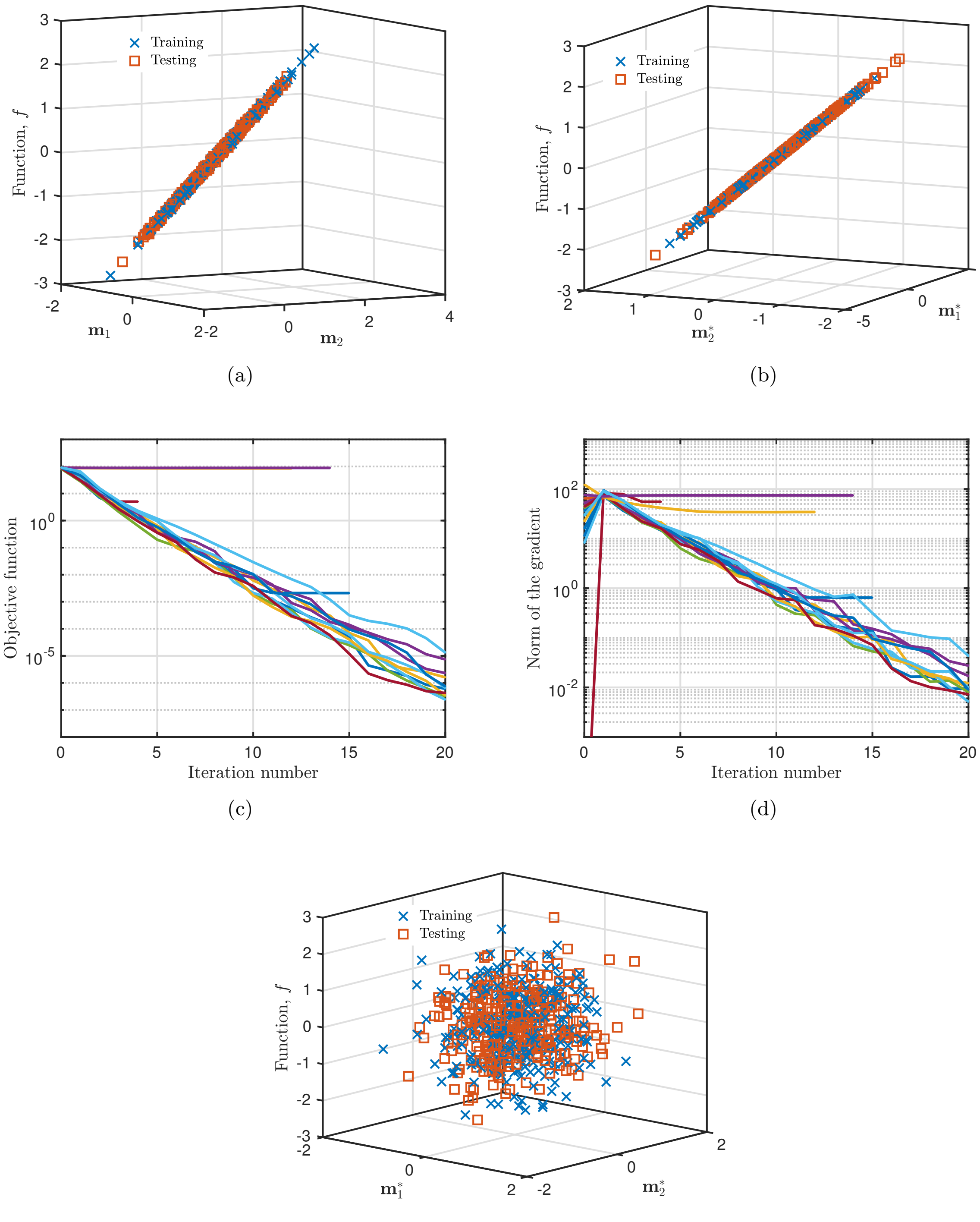
Figure 4. *Average probability of success values from* 20 *trials, for different training and testing samples for the case where* (a) $d = 10$*;* (b) $d = 100$ *for the analytical linear ridge problem. Darker colored circles denote a greater probability of success.*

pairs, starting with 50 in (a), (b) and 100 N_{test} and N_{train} samples each in (c), (d). As before, we examine a family of 20 random trials using the same testing and training input/output pairs, and then select the solution that yields the lowest value of the objective function. Once again, it is clear that our solution strategy is able to roughly identify the ridge structure associated with the problem.

4.3. Turbomachinery case study. In this subsection, we use the turbomachinery case study considered in [34] to compare the efficacy of our algorithm with a few sufficient dimension reduction techniques. We make two key claims in this subsection for this turbomachinery case study: (a) the standard deviation obtained from a Gaussian process may be used to make statements on the accuracy associated with the ridge approximation; and (b) our algorithm achieves a lower predictive variance compared to three well-known sufficient dimension reduction methods: SIR, SAVE, and CR. To begin, we briefly summarize the turbomachinery case presented here.

In their work investigating the use of active subspaces for learning turbomachinery aerodynamic pedigree rules of design, the authors of [34] parameterize an aeroengine fan blade with 25 design variables—five degrees of freedom defined at five spanwise locations. Once the geometry is generated, the mesh generation and flow solver codes described in [34] are used to solve the Reynolds average Navier–Stokes equations for the particular blade design; a postprocessing routine is then utilized for estimating the efficiency—our *quantity of interest*. In the aforementioned paper, the authors run a design of experiment (DOE) with 548 different designs for estimating the active subspaces via a global quadratic model, which required estimating 351 polynomial coefficients. In what follows, we demonstrate the efficacy of our algorithm with far fewer samples.

We randomly (uniformly) select $N = 300$ samples from the above DOE: input/output pairs ($\mathbf{x}_i \in \mathbb{R}^{25}$, $f_i \in \mathbb{R}$) for $i = 1, \ldots, N$. These samples are provided as inputs to SIR, SAVE, and



(a)

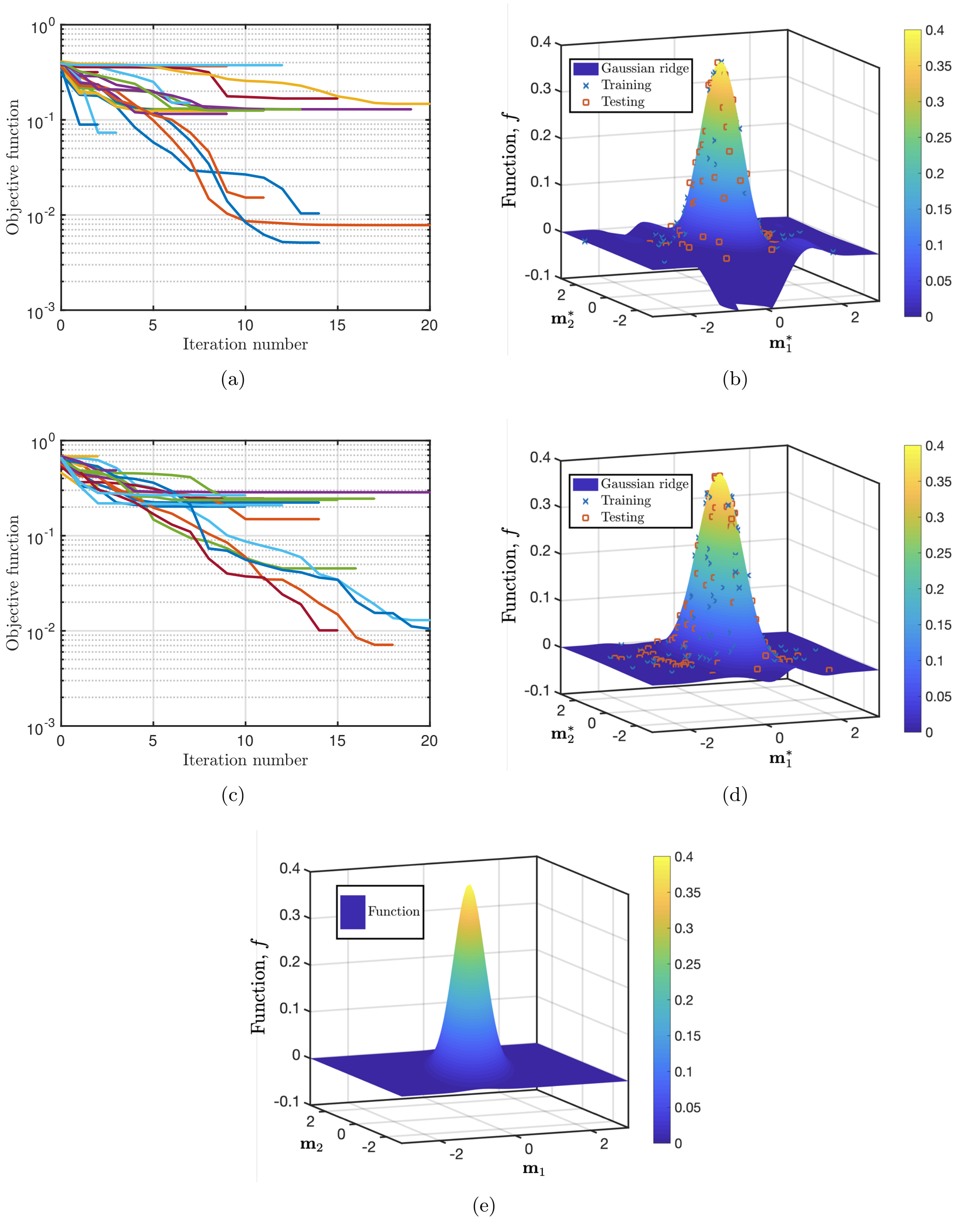
(b)

(c)

(d)

(e)

Figure 5. *Results of the analytical linear ridge problem with* $d = 100$*,* $N_{test} = 300$*, and* $N_{train} = 300$ *with* 20 *repetitions:* (a) *Sufficient summary plot on* $\boldsymbol{M}$*;* (b) *sufficient summary plot on* $\boldsymbol{M}^*$ *taken from the trial with the lowest value of the objective function;* (c) *objective function versus iterations;* (d) *gradient norm versus iterations; and* (e) *sufficient summary plot on* $\boldsymbol{M}^*$ *taken from the trial with a high value of the objective function.*



(a)

(b)

(c)

(d)

(e)

Figure 6. *Results of the bivariate normal distribution problem with $d = 5$ with the true function in* (e) *and approximations with* (a), (b) $N_{test} = 50$, $N_{train} = 50$ *and* (c), (d) $N_{test} = 100$, $N_{train} = 100$.

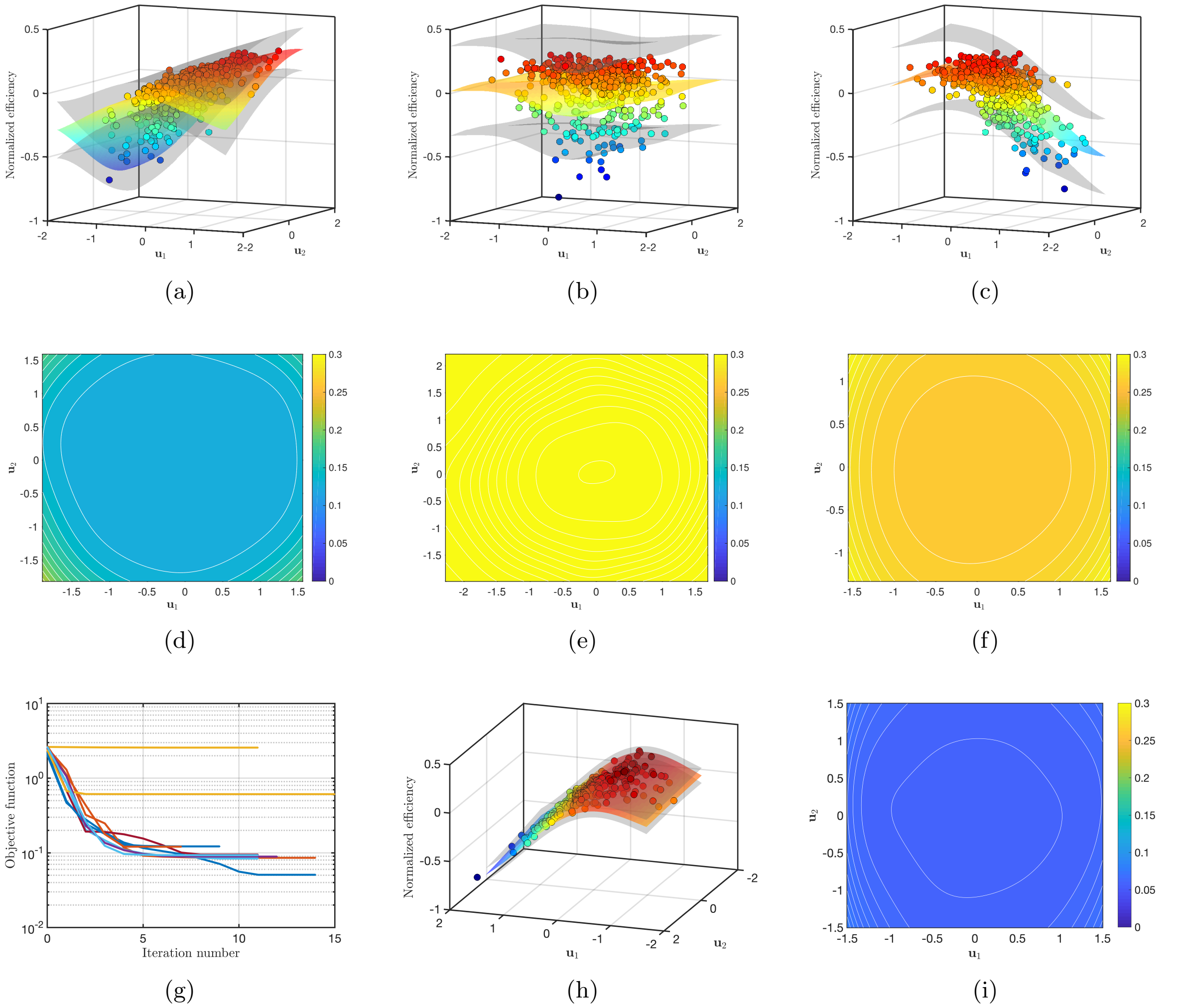


Figure 7. *Turbomachinery case study results for estimating the ridge subspace followed by Gaussian process regression for estimating the mean and 2σ response surfaces. Sufficient summary plots for SIR, SAVE, and CR are shown in* (a)–(c)*; corresponding 2σ contours are shown in* (d)–(f)*. Optimization convergence for* 20 *random trails for our algorithm is shown in* (g)*, sufficient summary plot in* (h)*, and the corresponding 2σ plot in* (i)*.*

CR for estimating a dimension-reducing subspace. The computed subspaces $\tilde{\boldsymbol{M}} \in \mathbb{R}^{d\times 2}$ are then used for generating sufficient summary plots and for fitting a Gaussian process regression response. To ensure parity, all methods use the same 300 samples.

To ascertain how appropriate the choice of a particular $\tilde{\boldsymbol{M}}$ is, we use all 548 DOE points in the sufficient summary plots and in the Gaussian process regression (even though the subspaces $\tilde{\boldsymbol{M}}$ are estimated with strictly 300 samples). Figures 7 (a)–(c) are the computed sufficient summary plots for SIR, SAVE, and CR. Here the vertical axis represents the normalized efficiency, while the horizontal axis for a particular design $\mathbf{x}_i$ is given by

$$\mathbf{u}_1 = \left(\tilde{\boldsymbol{M}}(:,1)\right)^T \mathbf{x} \quad \text{and} \quad \mathbf{u}_2 = \left(\tilde{\boldsymbol{M}}(:,2)\right)^T \mathbf{x}, \tag{49}$$

where the notation $(:,k)$ denotes the kth column of $\tilde{\boldsymbol{M}}$. Figures 7 (d)–(f) show contours of the 2σ values obtained from the Gaussian process response surface. It is clear from these plots that SIR yields lower 2σ values on average. SAVE and CR fail to identify any low-dimensional structure. In contrast, the result of our algorithm is shown in Figures 7 (g)–(i). The 2σ contours shown in Figure 7(i) are by far the lowest, and one can clearly see that all the data lies on a quadratic manifold. This result was obtained by selecting the subspace that yielded the lowest value of the objective function after 20 trials (see Figure 7(g)).

5. Conclusion.

In this paper we introduce a new algorithm for ridge function approximation tailored for subspace-based dimension reduction. Given point evaluations of a model, our algorithm approximates the ridge subspace $\boldsymbol{M}$ by minimizing a quadratic form of the function and its best ridge approximation. Throughout this paper, we assume our ridge function is the posterior mean of a Gaussian process. Our results for the examples considered demonstrate that (i) our algorithm is able to obtain near local minima recovery; and (ii) our algorithm offers better results compared with the three sufficient dimension reduction techniques considered in this paper.

Acknowledgment. The authors are grateful to Paul Constantine for insightful discussions.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2009.

[2] K. P. Adragni and R. D. Cook, *Sufficient dimension reduction and prediction in regression*, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 367 (2009), pp. 4385–4405, https://doi.org/10.1098/rsta.2009.0110.

[3] N. Boumal and P.-A. Absil, *Low-rank matrix completion via preconditioned optimization on the Grassmann manifold*, Linear Algebra Appl., 475 (2015), pp. 200–239.

[4] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *Manopt, a Matlab toolbox for optimization on manifolds*, J. Mach. Learn. Res., 15 (2014), pp. 1455–1459, http://jmlr.org/papers/v15/boumal14a.html.

[5] A. Cohen, I. Daubechies, R. DeVore, G. Kerkyacharian, and D. Picard, *Capturing ridge functions in high dimensions from point queries*, Constr. Approx., 35 (2012), pp. 225–243, https://doi.org/10.1007/s00365-011-9147-6.

[6] P. Constantine and D. Gleich, *Computing Active Subspaces with Monte Carlo*, preprint, https://arxiv.org/abs/1408.0545, 2014.

[7] P. G. Constantine, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM, Philadelphia, 2015, https://doi.org/10.1137/1.9781611973860.fm.

[8] P. G. Constantine, E. Dow, and Q. Wang, *Active subspace methods in theory and practice: Applications to Kriging surfaces*, SIAM J. Sci. Comput., 36 (2014), pp. A1500–A1524, https://doi.org/10.1137/130916138.

[9] P. G. Constantine, A. Eftekhari, J. Hokanson, and R. A. Ward, *A near-stationary subspace for ridge approximation*, Comput. Methods Appl. Mech. Engrg., 326 (2017), pp. 402–421, https://doi.org/10.1016/j.cma.2017.07.038.

[10] P. G. Constantine, B. Zaharatos, and M. Campanelli, *Discovering an active subspace in a single-diode solar cell model*, Stat. Anal. Data Min., 8 (2015), pp. 264–273, https://doi.org/10.1002/sam.11281.

[11] R. D. Cook, *Save: A method for dimension reduction and graphics in regression*, Comm. Statist. Theory Methods, 29 (2000), pp. 2109–2121, https://doi.org/10.1080/03610920008832598.

[12] R. D. Cook, *Rejoinder: Fisher Lecture: Dimension reduction in regression*, Statist. Sci., 22 (2007), pp. 40–43, https://doi.org/10.1214/088342307000000078.

[13] R. D. Cook, *Regression Graphics: Ideas for Studying Regressions through Graphics*, Wiley Ser. Probab. Statist. 482, John Wiley & Sons, New York, 2009.

[14] R. D. Cook and L. Ni, *Sufficient dimension reduction via inverse regression: A minimum discrepancy approach*, J. Amer. Statist. Assoc., 100 (2005), pp. 410–428, https://doi.org/10.1198/016214504000001501.

[15] P. Diaz, P. Constantine, K. Kalmbach, E. Jones, and S. Pankavich, *A modified SEIR model for the spread of Ebola in Western Africa and metrics for resource allocation*, Appl. Math. Comput., 324 (2018), pp. 141–155, https://doi.org/10.1016/j.amc.2017.11.039.

[16] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353, https://doi.org/10.1137/S0895479895290954.

[17] M. Fornasier, K. Schnass, and J. Vybiral, *Learning functions of few arbitrary linear parameters in high dimensions*, Found. Comput. Math., 12 (2012), pp. 229–262, https://doi.org/10.1007/s10208-012-9115-y.

[18] J. H. Friedman and W. Stuetzle, *Projection pursuit regression*, J. Amer. Statist. Assoc., 76 (1981), pp. 817–823.

[19] A. T. Glaws, P. G. Constantine, and R. D. Cook, *Inverse Regression for Ridge Recovery: A Data-driven Approach for Parameter Reduction in Computer Experiments*, preprint, https://arxiv.org/abs/1702.02227, 2017.

[20] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, 2013.

[21] Z. Grey and P. Constantine, *Active subspaces of airfoil shape parameterizations*, in Proceedings of the 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, 2017, https://doi.org/10.2514/6.2017-0507.

[22] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed., Springer Ser. Statist., Springer-Verlag, New York, 2009.

[23] B. Li and S. Wang, *On directional regression for dimension reduction*, J. Amer. Statist. Assoc., 102 (2007), pp. 997–1008, https://doi.org/10.1198/016214507000000536.

[24] K. C. Li, *Sliced inverse regression for dimension reduction*, J. Amer. Statist. Assoc., 86 (1991), pp. 316–327, https://doi.org/10.2307/2290563.

[25] X. Liu and S. Guillas, *Dimension reduction for Gaussian process emulation: An application to the influence of bathymetry on tsunami heights*, SIAM/ASA J. Uncertain. Quantif., 5 (2017), pp. 787–812, https://doi.org/10.1137/16M1090648.

[26] Y. Ma and L. Zhu, *A review on dimension reduction*, Int. Stat. Rev., 81 (2013), pp. 134–150, https://doi.org/10.1111/j.1751-5823.2012.00182.x.

[27] L. Magri, M. Bauerheim, F. Nicoud, and M. P. Juniper, *Stability analysis of thermo-acoustic nonlinear eigenproblems in annular combustors. Part* II. *Uncertainty quantification*, J. Comput. Phys., 325 (2016), pp. 411–421, https://doi.org/10.1016/j.jcp.2016.08.043.

[28] A. Pinkus, *Ridge Functions*, Cambridge Tracts in Math. 205, Cambridge University Press, Cambridge, UK, 2015, https://doi.org/10.1017/CBO9781316408124.

[29] E. Polak, *Optimization: Algorithms and Consistent Approximations*, Appl. Math. Sci. 124, Springer-Verlag, New York, 1997.

[30] F. Pukelsheim, *Optimal Design of Experiments*, Classics Appl. Math. 50, SIAM, Philadelphia, 2006, https://doi.org/10.1137/1.9780898719109.

[31] C. E. Rasmussen and H. Nickisch, *Gaussian processes for machine learning (GPML) toolbox*, J. Mach. Learn. Res., 11 (2010), pp. 3011–3015, http://www.jmlr.org/papers/v11/rasmussen10a.html.

[32] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, 1st ed., MIT Press, Cambridge, MA, 2006, http://www.gaussianprocess.org/gpml/.

[33] A. M. Samarov, *Exploring regression structure using nonparametric functional estimation*, J. Amer. Statist. Assoc., 88 (1993), pp. 836–847, https://doi.org/10.2307/2290772.

[34] P. Seshadri, S. Shahpar, P. Constantine, G. Parks, and M. Adams, *Turbomachinery active subspace performance maps*, J. Turbomach., 140 (2018), 041003, https://doi.org/10.1115/1.4038839.

[35] R. Tripathy, I. Bilionis, and M. Gonzalez, *Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation*, J. Comput. Phys., 321 (2016), pp. 191–223, https://doi.org/10.1016/j.jcp.2016.05.039.

[36] H. Tyagi and V. Cevher, *Learning non-parametric basis independent models from point queries via low-rank methods*, Appl. Comput. Harmon. Anal., 37 (2014), pp. 389–412, https://doi.org/10.1016/j.acha.2014.01.002.

[37] L.-P. Zhu and L.-X. Zhu, *Dimension reduction for conditional variance in regressions*, Statist. Sinica, 19 (2009), pp. 869–883, http://www.jstor.org/stable/24308860.