A Local Approach to Parameter Space Reduction for Regression and Classification Tasks

Francesco Romor[1] · Marco Tezzele[1] · Gianluigi Rozza[1]

Abstract

Parameter space reduction has been proved to be a crucial tool to speed-up the execution of many numerical tasks such as optimization, inverse problems, sensitivity analysis, and surrogate models' design, especially when in presence of high-dimensional parametrized systems. In this work we propose a new method called local active subspaces (LAS), which explores the synergies of active subspaces with supervised clustering techniques in order to carry out a more efficient dimension reduction in the parameter space. The clustering is performed without losing the input–output relations by introducing a distance metric induced by the global active subspace. We present two possible clustering algorithms: K-medoids and a hierarchical top–down approach, which is able to impose a variety of subdivision criteria specifically tailored for parameter space reduction tasks. This method is particularly useful for the community working on surrogate modelling. Frequently, the parameter space presents subdomains where the objective function of interest varies less on average along different directions. So, it could be approximated more accurately if restricted to those subdomains and studied separately. We tested the new method over several numerical experiments of increasing complexity, we show how to deal with vectorial outputs, and how to classify the different regions with respect to the LAS dimension. Employing this classification technique as a preprocessing step in the parameter space, or output space in case of vectorial outputs, brings remarkable results for the purpose of surrogate modelling.

✉ Gianluigi Rozza
gianluigi.rozza@sissa.it

Francesco Romor
francesco.romor@sissa.it

Marco Tezzele
marco.tezzele@sissa.it

[1] Mathematics Area, mathLab, SISSA, Via Bonomea 265, 34136 Trieste, Italy

   🍃 Springer

1 Introduction

Parameter space reduction [1, 2] is a rapidly growing field of interest which plays a key role in fighting the curse of dimensionality. The need of reducing the number of design inputs is particularly important in engineering for advanced CFD simulations to model complex phenomena, especially in the broader context of model order reduction [3–6] and industrial numerical pipelines [7–9].

Active subspaces [1] is one of the most used techniques for linear reduction in input spaces. It has been proved useful in many numerical tasks such as regression, using a multi-fidelity data fusion approach with a surrogate model built on top of the AS as low-fidelity model [10], shape optimization [11–13] and a coupling with the genetic algorithm to enhance its performance [14, 15], inverse problems [16], and uncertainty quantification [17]. It has also been used to enhance classical model order reduction techniques such as POD-Galerkin [18], and POD with interpolation [19, 20]. Other attempts towards nonlinear parameter space reduction have been proposed recently: kernel-based active subspaces [21], nonlinear level-set learning [22], and active manifold [23] are the most promising. In [24], instead, they project the input parameters onto a low-dimensional subspace spanned by the eigenvectors of the Hessian corresponding to its dominating eigenvalues.

In this work we propose a new local approach for parameter space dimensionality reduction for both regression and classification tasks, called local active subspaces (LAS). In our work we do not simply apply a clustering technique to preprocess the input data, we propose a supervised metric induced by the presence of a global active subspace. The directions individuated by local active subspaces are locally linear, and they better capture the latent manifold of the target function.

From a wider point of view, there is an analogy between local parameter space reduction and local model order reduction. With the latter, we mean both a spatial domain decomposition approach for model order reduction of parametric PDEs in a spatial domain $\Omega \subset \mathbb{R}^d$ and a local reduction approach in the parameter space. As representatives methods for the first paradigm we report the reduced basis element method [25], which combines the reduced basis method in each subdomains with a mortar type method at the interfaces, and more in general domain decomposition methods applied to model order reduction. For the second approach we cite the interpolation method in the Grassmannian manifold of the reduced subspaces [26]; in particular in [27] the K-medoids clustering algorithm with Grassmann metric is applied to the discrete Grassmann manifold of the training snapshots as a step to perform local model order reduction. With this work we fill the gap in the literature regarding localization methods in the context of parameter space reduction.

Other methods have been developed in the last years exploiting the localization idea. We mention localized slice inverse regression (LSIR) [28] which uses local information of the slices for supervised regression and semi-supervised classification. LSIR improves local discriminant information [29] and local Fischer discriminant analysis [30] with more efficient computations for classification problems. The main difference between slice inverse regression (SIR) [31] and AS is in the construction of the projection matrix. While SIR needs the elliptic assumption, AS exploits the gradients of the function of interest with respect to the input parameters. Recently, a new work on the subject was disclosed [32]. Here we emphasize the differences and the original contributions of our work: (1) we implemented hierarchical top–down clustering applying K-medoids with a new metric that includes the gradient information through the active subspace. In [32] they employed hierarchical bottom-up clustering with unweighted average linkage and a distance obtained as a weighted sum of

the Euclidean distance of the inputs and the cosine of the angle between the corresponding gradients; (2) we included for vector-valued objective functions and answered questions about the employment of the new method to decrease the ridge approximation error with respect to a global approach; (3) we also focused on classification algorithms and we devised a method to classify the inputs based on the local active subspace dimension with different techniques, including the use of the Grassmannian metric; (4) our benchmarks include vector-valued objective functions from computational fluid dynamics. We also show that clustering the outputs with our classification algorithms as pre-processing step leads to more efficient surrogate models.

This work is organized as follows: in Sect. 2 we briefly review the active subspaces method, in Sect. 3 we introduce the clustering algorithms used and the supervised distance metric based on the presence of a global active subspace, focusing on the construction of response surfaces and providing theoretical considerations. In Sect. 4 we present the algorithms to exploit LAS for classification. We provide extensive numerical results in Sect. 5 from simple illustrative bidimensional dataset to high-dimensional scalar and vector-valued functions. Finally, in Sect. 6 we draw conclusions and future perspectives.

2 Active Subspaces for Parameter Space Reduction

Active subspaces (AS) [1] are usually employed as dimension reduction method to unveil a lower dimensional structure of a function of interest f, or provide a global sensitivity measure not necessarily aligned with the coordinate axes [33]. Through spectral considerations about the second moment matrix of f, the AS identify a set of linear combinations of the input parameters along which f varies the most on average.

We make some general assumptions about the inputs and function of interest [1, 33, 34]. Let us introduce the inputs as an absolutely continuous random vector $\mathbf{X}$ with values in $\mathbb{R}^n$ and probability distribution $\boldsymbol{\mu}$. We represent with $\mathcal{X} \subset \mathbb{R}^n$ the support of $\boldsymbol{\mu}$ and as such our parameter space. We want to compute the active subspace of a real-valued function $f : (\mathcal{X}, \mathcal{B}(\mathbb{R}^n), \boldsymbol{\mu}) \to \mathbb{R}$, where $\mathcal{B}(\mathbb{R}^n)$ is the Borel σ-algebra of $\mathbb{R}^n$. We denote with $\mathbf{x} \in \mathcal{X}$ an element in the space of parameters and with $\{\mathbf{x}_i\}_i$ a set of realizations of $\mathbf{X}$.

An extension to vector-valued functions has been presented in [34] and extended for kernel-based AS in [21]. Even if in this section we focus only on scalar functions, the following considerations can be carried over to the multivariate case without too much effort.

Let Σ be the second moment matrix of ∇f defined as

$$\Sigma := \mathbb{E}_{\boldsymbol{\mu}} \left[\nabla_{\mathbf{x}} f \, \nabla_{\mathbf{x}} f^T \right] = \int (\nabla_{\mathbf{x}} f)(\nabla_{\mathbf{x}} f)^T \, d\boldsymbol{\mu}, \tag{1}$$

where $\mathbb{E}_{\boldsymbol{\mu}}$ denotes the expected value with respect to $\boldsymbol{\mu}$, and $\nabla_{\mathbf{x}} f = \nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial \mathbf{x}_1}, \ldots, \frac{\partial f}{\partial \mathbf{x}_n} \right]^T$ is the column vector of partial derivatives of f. Its real eigenvalue decomposition reads $\Sigma = \mathbf{W} \Lambda \mathbf{W}^T$. We can retain the most energetic eigenpairs by looking at the spectral decay of the matrix Σ. The number r of eigenpairs we select is the active subspace dimension, and the span of the corresponding eigenvectors defines the active subspace. The partition is the following

$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \qquad \mathbf{W} = [\mathbf{W}_1 \quad \mathbf{W}_2], \tag{2}$$

where $\Lambda_1 = \text{diag}(\lambda_1, \ldots, \lambda_r)$, and $\mathbf{W}_1$ contains the first r eigenvectors arranged by columns. With this matrix we can project the input parameters onto the active subspace, and its orthogonal complement, that is the inactive subspace, as follows:

$$\mathbf{Y} = P_r(\mathbf{X}) = \mathbf{W}_1 \mathbf{W}_1^T \mathbf{X} \in \mathbb{R}^n, \qquad \mathbf{Z} = (I - P_r)(\mathbf{X}) = \mathbf{W}_2 \mathbf{W}_2^T \mathbf{X} \in \mathbb{R}^n, \qquad (3)$$

with $P_r : \mathbb{R}^n \to \mathbb{R}^n$ the linear projection operator $P_r := \mathbf{W}_1 \mathbf{W}_1^T$. The selection of the active subspace dimension r can be set a priori, or by looking at the presence of a spectral gap [1], or by imposing a cumulative energy threshold for the eigenvalues.

We will consider the problem of ridge approximation [35] in our applications. The AS are, in fact, the minimizers of an upper bound of the ridge approximation error.

Definition 1 (Ridge approximation) Given $r \in \mathbb{N}$, $r \ll n$ and a tolerance $\epsilon \geq 0$, find the profile function $h : (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n), \boldsymbol{\mu}) \to \mathbb{R}$ and the r-rank projection $P_r : \mathbb{R}^n \to \mathbb{R}^n$ such that the following upper bound on the ridge approximation error is satisfied

$$\mathbb{E}_{\boldsymbol{\mu}}[\|f - h \circ P_r\|_2^2] \leq \epsilon^2, \qquad (4)$$

where $\|\cdot\|_2$ is the L^2-norm of $\mathbb{R}$.

For a fixed projection P_r the optimal profile $\tilde{h}$ is given by the conditioned random variable $\mathbb{E}_{\boldsymbol{\mu}}[f|P_r]$. Under the additional assumptions on the probability distribution $\boldsymbol{\mu}$, reported in Sect. 6.1 of the Appendix, the AS can indeed be defined as a minimizer of an upper bound of the ridge approximation error [1, 21, 34, 36]. The proof is a direct consequence of the Poincaré inequality and standard properties of eigenspaces, and for this specific version of the theorem it can be found in [36].

Theorem 1 (*Definition of AS through ridge approximation*) *The solution P_r of the ridge approximation problem in Definition 1, with optimal profile $\tilde{h} = \mathbb{E}_{\boldsymbol{\mu}}[f|P_r]$, is the orthogonal projector to the eigenspace of the first r-eigenvalues of Σ ordered by magnitude*

$$\Sigma v_i = \lambda_i v_i \qquad \forall i \in \{1, \ldots, n\}, \qquad \tilde{P}_r = \sum_{j=1}^{r} v_j \otimes v_j,$$

with $r \in \mathbb{N}$ chosen such that

$$\mathbb{E}_{\boldsymbol{\mu}} \left[\|f - \tilde{h}\|_2^2 \right] \leq C(C_P, \tau) \operatorname*{argmin}_{\substack{P^2 = P, P = P^T, \\ rank(P) = r}} \mathbb{E}_{\boldsymbol{\mu}}[\|(Id - P)\nabla f\|^2]^{\frac{1}{1+\tau}}$$

$$\leq C(C_P, \tau) \left(\sum_{i=r+1}^{m} \lambda_i \right)^{\frac{1}{1+\tau}} \leq \epsilon^2.$$

with $C(C_P, \tau)$ a constant depending on $\tau > 0$ related to the choice of $\boldsymbol{\mu}$ and on the Poincaré constant C_P, and $\tilde{h} = \mathbb{E}_{\boldsymbol{\mu}}[f|\sigma(P_r)]$ is the conditional expectation of f given the σ-algebra generated by the random variable $P_r \circ \mathbf{X}$.

To ease the notation, in the following we will consider only the first three classes of probability distribution in the assumptions of Sect. 6.1 in the "Appendix", such that $\tau = 0$.

3 Localized Parameter Space Reduction

Sometimes we do not have a priori knowledge about the target function's behaviour in a particular parameter space region. This could lead to a poor selection of the parameters range, hugely affecting optimization tasks. In these cases, a preprocessing of the data using a clustering technique could be highly beneficial. With a clustering of the input parameters, we can treat each subregion separately, and thus capture more accurately the target function's variability. This is always true for any function of interest, but for functions with global lower intrinsic dimensionality we can exploit such structure to enhance the clustering. To this end, we propose a new distance metric for K-medoids and hierarchical top–down clustering methods which exploits the global active subspace of the target function. By applying AS on each cluster we find the optimal rotation of the corresponding subregion of the input domain, which aligns the data along the active subspace of a given dimension.

In this section, we make some theoretical considerations regarding ridge approximation applied to partitions of the parameter space and review three clustering methods [37]: K-means, K-medoids, and hierarchical top–down clustering [38, 39]. We are going to use K-means as the baseline since the input parameter space is assumed to be a hyperrectangle. This assumption covers the majority of the practical test cases in the reduced order modeling community.

3.1 Ridge Approximation with Clustering and Active Subspaces

Regardless of the choice of clustering algorithm, given a partition of the parameter space we want to perform ridge approximation with AS in each subdomain. We will introduce some definitions and make some remarks to clarify the setting. The function of interest f represents scalar outputs, but the following statements can be extended to vector-valued outputs as well.

Definition 2 (Local ridge approximation with active subspaces) Given a partition of the domain $\mathcal{P} := \{S_i\}_{i \in \{1,\dots,d\}}$ and a map $r : \mathcal{P} \to \{1,\dots,n_r\}$, $n_r \ll n$ representing the local reduced dimension, the local ridge approximation with active subspaces of $(f, \boldsymbol{\mu})$ is the function $R_{AS}(r, f, \boldsymbol{\mu}) : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ that is defined locally for every $S_i \in \mathcal{P}$ as

$$g|_{S_i} = \mathbb{E}_{\boldsymbol{\mu}_i}[f | P_{r(S_i),i}], \tag{5}$$

where $\boldsymbol{\mu}_i := (1/\boldsymbol{\mu}(S_i)) \cdot \boldsymbol{\mu}|_{S_i} \in \mathbb{R}^n$, and $P_{r(S_i),i} : S_i \subset \mathbb{R}^n \to \mathbb{R}^n$ is the orthogonal projector with rank r that satisfies the minimization problem:

$$P_{r(S_i),i} = \underset{\substack{P^2 = P, P = P^T, \\ \text{rank}(P) = r}}{\text{argmin}} \mathbb{E}_{\boldsymbol{\mu}_i} \|(Id - P)\nabla f\|^2. \tag{6}$$

With this definition we can state the problem of local ridge approximation with active subspaces.

Problem 1 (Minimizers $(\mathcal{P}, r)$ of the ridge approximation error) Find the partition $\mathcal{P}$ of the domain $\mathcal{X} \subset \mathbb{R}^n$ and the local reduced dimension map $r : \mathcal{P} \to \{1, \dots, n_r\}$, $n_r \ll n$, such that the L^2-error between the objective function f and its local ridge approximation with active subspaces is minimized.

$$\mathbb{E}_{\boldsymbol{\mu}} \left[\|f - R_{AS}(r, f)\|^2 \right] = \sum_{S_i \in \mathcal{P}} \mathbb{E}_{\boldsymbol{\mu}} \left[\|f|_{S_i} - \mathbb{E}_{\boldsymbol{\mu}_i}[f | P_{r(S_i),i}]\|^2 \right]. \tag{7}$$

Assuming that the subspace Poincaré inequality [36] is valid also for $(f, \boldsymbol{\mu})$ restricted to the elements of the partition $\mathcal{P}$, a straightforward bound is obtained by applying the Poincaré inequality for every element of the partition

$$\mathbb{E}_{\boldsymbol{\mu}} \left[\| f - R_{\text{AS}}(r, f) \|^2 \right] = \sum_{S_i \in \mathcal{P}} \mathbb{E}_{\boldsymbol{\mu}} \left[\| f|_{S_i} - \mathbb{E}_{\boldsymbol{\mu}_i} [f | P_{r(S_i), S_i}] \|^2 \right]$$

$$\lesssim \sum_{S_i \in \mathcal{P}} \mathbb{E}_{\boldsymbol{\mu}} \left[\| (Id - P_{r(S_i), i}) \nabla f \|^2 \right].$$

To obtain the previous upper bound, we made an assumption about the Poincaré subspace inequality that in general is not satisfied by any probability measure $\boldsymbol{\mu}$ chosen: the assumptions on the probability distributions $\{\mu_i\}_{i=1}^d$ in Sect. 6.1 of the Appendix have to be satisfied at each subdomain $\{S_i\}_{i=1}^d$.

For the moment we will consider the local reduced dimension map r constant and, in general, the codomain of r is a subset of $\{1, \ldots, n_r\}$, $n_r \ll n$.

The previous bound suggests that a good indicator for refinement could be represented by the sum of the residual eigenvalues $\{\lambda_{S_i, j}\}_{j=r_{S_i}}^m$ of the local correlation matrices, for every $S_i \in \mathcal{P}$:

$$\mathbb{E}_{\boldsymbol{\mu}} \left[\| f - R_{\text{AS}}(r, f) \|^2 \right] \lesssim \sum_{S_i \in \mathcal{P}} \sum_{j=r(S_i)+1}^m \lambda_{S_i, j}.$$

We also have the following immediate result that hints to indefinitely many successive refinements to lower the L^2-error ridge approximation error.

Remark 1 (Relationships between the upper bounds of consecutive refinements) Considering the sum over the number of refined clusters $cl \in \{1, \ldots, d\}$ we have that

$$\int_{\mathcal{X}} \| (Id - P_r) \nabla f \|^2 \, d\boldsymbol{\mu} = \sum_{cl=1}^d \int_{S_{cl} \subset \mathcal{X}} \| (Id - P_r) \nabla f \|^2 \, d\boldsymbol{\mu}$$

$$\geq \sum_{cl=1}^d \int_{S_{cl} \subset \mathcal{X}} \| (Id - P_{r, cl}) \nabla f \|^2 \, d\boldsymbol{\mu}, \quad (8)$$

since the projectors $\{P_{r, cl}\}_{cl \in \{1, \ldots, d\}}$ are the minimizers of

$$P_{r, cl} = \underset{\substack{P^2 = P, P = P^T, \\ \text{rank}(P) = r}}{\text{argmin}} \int_{S_{cl} \subset \mathcal{X}} \| (Id - P) \nabla f \|^2 \, d\boldsymbol{\mu}. \quad (9)$$

The RHS of Eq. 8 can be used as indicator for refinement. We remark that since the refinements increase the decay of the eigenvalues in the RHS of Eq. 8, the choice of the dimension of the active subspace may be shifted towards lower values to achieve further dimension reduction for the same accuracy, as we are going to show in the numerical experiments, in Sect. 5.

Unfortunately, the minimizers of the ridge approximation error and of the upper bound are not generally the same:

$$\underset{\{P_{r(S_i), i}\}_{S_i \in \mathcal{P}}}{\text{argmin}} \mathbb{E}_{\boldsymbol{\mu}} \left[\| f - R_{\text{AS}}(r, f) \|^2 \right] \neq \underset{\{P_{r(S_i), S_i}\}_{S_i \in \mathcal{P}}}{\text{argmin}} \sum_{S_i \in \mathcal{P}} \mathbb{E}_{\boldsymbol{\mu}_i} \left[\| (Id - P_{r(S_i), i}) \nabla f \|^2 \right].$$

There is a counterexample for the non localized case in [34]. We start from this counterexample to show that in general the L^2-error of the local ridge approximation does not decrease

between consequent refinements, even if the indicator from the RHS of Eq. 8 does, as stated in the previous remark.

Corollary 1 *(Counterexample for indefinite refinement as optimal clustering criterion) Let* $\mathcal{P} = \{A, B, C\}$ *be a partition of* $\mathcal{X} = [-1, 1]^2$ *such that* $A = [-1, \epsilon] \times [-1, 1]$, $B = [-\epsilon, \epsilon] \times [-1, 1]$, *and* $C = [\epsilon, 1] \times [-1, 1]$. *Let* $\boldsymbol{\mu}$ *be the uniform probability distribution on* $\mathcal{X}$. *The objective function we want to approximate is*

$$f : \mathcal{X} \subset \mathbb{R}^2 \to \mathbb{R}, \quad f = \begin{cases} x_1 + \epsilon, & \mathbf{x} \in A, \\ x_1(x_1 + \epsilon)(x_1 - \epsilon)\cos(\omega x_2), & \mathbf{x} \in B, \\ x_1 - \epsilon, & \mathbf{x} \in C, \end{cases} \quad (10)$$

with local reduced dimension map $r(A) = r(B) = r(C) = 1$. *There exist* $\epsilon > 0, \omega > 0$, *such that*

$$\mathbb{E}_{\boldsymbol{\mu}}\left[\|f - R_{AS}(r, f, \boldsymbol{\mu})\|^2\right] \geq \mathbb{E}_{\boldsymbol{\mu}}\left[\|f - \mathbb{E}_{\boldsymbol{\mu}}\left[f|P_{1,\mathcal{X}}\right]\|^2\right],$$

where $P_{1,\mathcal{X}}$ *is the optimal projector on the whole domain* $\mathcal{X}$ *with one-dimensional active subspace.*

Proof The proof is reported in Sect. 6.3 of the Appendix. □

The heuristics behind the previous proof rests on the fact that ridge approximation with active subspaces performs poorly when the objective function has a high variation. The counterexample is valid whenever the global projector $P_{1,\mathcal{X}}$ is the minimizer of a local L^2 ridge approximation error for which the minimizer of the gradient-based indicator in Eq. 8 does not coincide. This leaves us with an indicator in Eq. 8 that does not guarantee a non increasing L^2-error decay for subsequent refinements, but is nonetheless useful in practice.

We conclude the section with some remarks about the response surface design through the ridge approximation with active subspaces.

Remark 2 (Approximation of the optimal profile) In practice we do not consider the optimal profile $h(\mathbf{y}) = \mathbb{E}_{\boldsymbol{\mu}}[f|\sigma(P_r)](\mathbf{y})$ but we employ the approximation $h(\mathbf{y}) = f(\mathbf{y}) = f(P_r\mathbf{x})$. The reason lies on the fact that to approximate the optimal profile at the values $\{\mathbf{y}_i\}_i$, additional samples from the conditional distribution $p(z|\mathbf{y}_i = P_r\mathbf{x})$ must be obtained; even if the accuracy of the ridge approximation could benefit from it, this is not always possible in practice because of the difficulty to sample from the conditional distribution or because of computational budget constraints.

If the data is split in training, validation, and test set, the local R^2 score on the validation set can be used as indicator for refinement.

Remark 3 (Estimator based on local R^2 scores) The R^2 score of a single cluster can be written with respect to the R^2 scores $\{R_l^2\}_{l \in \{1,...,d\}}$ relative to the clusters of the subsequent refinement. Let the sum be over the refinement clusters $l \in \{1, \ldots, d\}$, we have

$$R^2 = 1 - \frac{\mathbb{E}[\|f - \mathbb{E}[f|P_r]\|^2]}{\text{Var}(f)} = 1 - \sum_{l=1}^{d} \frac{\mathbb{E}[\|f|_{S_l} - \mathbb{E}[f|P_{r,l}]\|^2]}{\text{Var}(f)}$$

$$= 1 - \sum_{l=1}^{d} \frac{\text{Var}(f|_{S_l})}{\text{Var}(f)} \cdot \frac{\mathbb{E}[\|f|_{S_l} - \mathbb{E}[f|P_{r,l}]\|^2]}{\text{Var}(f|_{S_l})} = 1 - \sum_{l=1}^{d} \frac{\text{Var}(f|_{S_l})}{\text{Var}(f)} \cdot (1 - R_l^2), \quad (11)$$

which, substituting with the empirical variance, becomes

$$R_{\text{emp}}^2 = 1 - \sum_{l=1}^{d} \frac{\text{Var}_{\text{emp}}(f|_{S_l})}{\text{Var}_{\text{emp}}(f)} \cdot (1 - R_{\text{emp};l}^2) \cdot \frac{N_l - 1}{N - 1}, \tag{12}$$

where $R_{\text{emp};l}^2$ is the empirical local R^2 score relative to cluster number l. The definition can be extended for component-wise vector-valued objective functions f. The numerical results shown in Sect. 5 consider the mean R^2 score along the components when the output is vectorial.

In practice every expectation is approximated with simple Monte Carlo, and without the number of training samples increasing, the confidence on the approximation is lower and lower, the more the domain is refined. This is taken into consideration while clustering, fixing a minimum number of samples per cluster for example.

The Sect. 6.2 in the Appendix clarifies the link between the number of Monte Carlo samples, the numerical method chosen for the discretization of the integral $\mathbb{E}_{\boldsymbol{\mu}}[\nabla f \otimes \nabla f]$, and the approximation of the active subspace. For example for deterministic models, one could employ the more efficient Sobol' sequence or a Latin hypercube sampling; if f is more regular and the parameter space dimension is not too high one could employ tensor product Gauss quadrature rule. See for example [33].

Before introducing the clustering algorithms we will employ, we specify that the partition $\mathcal{P} = \{S_i\}_{i \in \{1,...,d\}}$ is defined by the decision boundaries of the clustering algorithm chosen.

3.2 K-means Clustering

We recall the K-means clustering algorithm. Let $\{x_i\}_{i=1}^{N}$ be a set of N samples in $\mathbb{R}^{N_F}$, where N_F denotes the number of features. The K-means algorithm divides this set into K disjoint clusters $S = \{S_j\}_{j=1}^{K}$, with $S_l \cap S_m = \emptyset$ for $1 \leq l, m \leq K$ and $l \neq m$. The partitioning quality is assessed by a function which aims for high intracluster similarity and low intercluster similarity. For K-means this is done by minimizing the total within-cluster sum-of-squares criterion W_T, which reads as

$$W_T(S) := \sum_{j=1}^{K} W(S_j) = \sum_{j=1}^{K} \sum_{x_i \in S_j} \|x_i - c_j\|_{L^2}^2, \tag{13}$$

where c_j is the centroid describing the cluster S_j. A centroid of a cluster is defined as the mean of all the points included in that cluster. This means that the centroids are, in general, different from the samples x_i.

K-means is sensitive to outliers, since they can distort the mean value of a cluster and thus affecting the assignment of the rest of the data.

3.3 K-medoids Clustering with Active Subspaces-Based Metric

In order to overcome some limitations of the K-means algorithm, such as sensitivity to outliers, we can use K-medoids clustering technique [38, 40–42]. It uses an actual sample as cluster representative (i.e. medoid) instead of the mean of the samples within the cluster.

Following the notation introduced in the previous section, let m_j be the medoid describing the cluster S_j. The partitioning method is performed by minimizing the sum of the dissimilarities between the samples within a cluster and the corresponding medoid. To this end, an absolute-error criterion E is used, which reads as

$$E(S) := \sum_{j=1}^{K} E(S_j) = \sum_{j=1}^{K} \sum_{x_i \in S_j} \|x_i - m_j\|. \tag{14}$$

By looking at the formula above it is clear that the use of a data point to represent each cluster's center allows the use of any distance metric for clustering. We remark that the choice of the Euclidean distance does not produce the same results as K-means because of the different references representing the clusters.

We propose a new supervised distance metric inspired by the global active subspace of the function f we want to approximate. We define a scaled L^2 norm using the eigenpairs of the second moment matrix of ∇f, which is the matrix from which we calculate the global active subspace:

$$\|x_i - x_j\|_\Lambda = \sqrt{(x_i - x_j)^T \mathbf{W} \Lambda \mathbf{W}^T (x_i - x_j)}, \tag{15}$$

where Λ stands for the diagonal matrix with entries the eigenvalues of Eq. (1), and $\mathbf{W}$ is the eigenvectors matrix from the decomposition of the covariance matrix. As we are going to show in Sect. 5 this new metric allows a better partitioning both for regression and classification tasks by exploiting both global and local informations. For insights about the heuristics behind it, we refer to remark 5.

To actually find the medoids the partitioning around medoids (PAM) algorithm [38] is used. PAM uses a greedy approach after the initial selection of the medoids, also called representative objects. The medoids are changed with a non-representative object, i.e. one of the remaining samples, if it would improve the clustering quality. This iterative process of replacing the medoids by other objects continues until the quality of the resulting clustering cannot be improved by any replacement. Algorithm 1 presents this approach with pseudo-code.

Algorithm 1 K-medoids algorithm with AS metric.

    Input: set of samples $\{x_i\}_{i=1}^{N} \in \mathbb{R}^{N_F}$
               number of clusters K
               distance metric d defined in Eq. (15)
    Output: set of clusters $S = \{S_j\}_{j=1}^{K}$
1: select initial cluster medoids
2: **repeat**
3:      assign each sample to its closest medoid using the distance metric d
4:      randomly select K non-representative objects
5:      swap the medoids with the new selected objects by minimizing Eq. (14)
6: **until** clustering quality converges

3.4 Hierarchical Top–Down Clustering

In this section, we present hierarchical top–down clustering [38, 39], and exploit the additional information from the active subspace, as done for K-medoids. In the following sections, we refer to this technique with the acronym HAS.

In top–down hierarchical clustering, at each iteration the considered clusters, starting from the whole dataset, are split further and further based on some refinement criterion, until convergence. A nice feature of hierarchical clustering algorithms, with respect to K-means

and K-medoids, is that the number of clusters can be omitted. Moreover, by stopping at the first refinement and forcing the total number of clusters to be the maximum number of clusters specified, HAS can be seen as a generalization of the previous methods: for this reason, we wanted to make the implementation consistent with K-means and K-medoids with AS induced metric as close as possible, as shown in the numerical results in Sect. 5.

Pushing further the potential of clustering algorithms applied to local dimension reduction in the parameter space, HAS is a versatile clustering method that takes into account the variability of the AS dimension along the parameter space. The price paid for this is the overhead represented by the tuning of some hyper-parameters introduced later.

A schematic representation of the algorithm of top–down clustering is reported in Algorithm 2. The design is straightforward and it employs a tree data structure that assigns at each node a possible clustering of the whole dataset: consequent refinements are represented by children nodes down until the leaves of the tree, that represent the final clusters.

Remark 4 (Normalization of the clusters at each refinement iteration) Each cluster, at every refinement step, is normalized uniformly along dimensions onto the hyper-cube domain $[-1, 1]^n$, even if the subdomain identified by the cluster is not a hyperrectangle. Another possible choice for normalization is standardization, centering the samples with their mean and dividing them by their standard deviation.

Algorithm 2 Hierarchical top–down algorithm.

Input: set of samples $S = \{x_i\}_{i=1}^N \in \mathbb{R}^{N_F}$
       maximum number of clusters K
       range of number of children $\{n_{min}^{child}, n_{max}^{child}\}$
       minimum number of elements in a cluster n_{el}
       indicator for refinement I
       distance metric d
       minimum and maximum AS dimensions r_{min}, r_{max}
       score tolerance ϵ
Output: refinement tree T
1: add the initial cluster S to FIFO queue $q = \{S\}$
2: **while** $q \neq \varnothing$ **do**
3:    take S_j, the first element from queue q
4:    apply the refinement function in algorithm 3 to S_j to get $\{S_i\}_i$
5:    add $\{S_i\}_i$ to the queue q
6:    **if** the score tolerance ϵ is reached **or** other constraints are violated **then**
7:        **break**
8:    **end if**
9: **end while**

The procedure depends on many parameters that have to be tuned for the specific case or depend a priori on the application considered: the maximum number of clusters (K), the minimum and maximum number of children nodes (n_{min}^{child}, n_{max}^{child}), the tolerance for the score on the whole domain (ϵ), the minimun and maximum dimension of the active subspace (r_{min}, r_{max}), and the minimum number of elements (n_{el}) of each cluster (usually $n_{el} > r$, where r is the local AS dimension).

More importantly the method is versatile for the choice of clustering criterion, indicator for refinement (I), distance metric (d, from Eq. (15)) and regression method. In the following sections we consider K-means and K-medoids with the active subspaces distance as clustering criterion (see Sect. 3.3), but other clustering algorithms could in principle be applied at each refinement.

Algorithm 3 Refinement function.

Input: cluster $S = \{x_i\}_{i=1}^N \in \mathbb{R}^{N_F}$
        number of clusters per tree refinement level K
        range of number of children $\{n_{min}^{child}, n_{max}^{child}\}$
        minmum number of elements in a cluster n_{el}
        indicator for refinement I
        distance metric d
        minimum and maximum AS dimensions r_{min}, r_{max}
Output: $\{S_j\}_{j=1}^{n^{child}}$, the children of cluster S

1: set best score to $b = 0$
2: **for each** n^{child} from n_{min}^{child} to n_{max}^{child} **do**
3:    apply the chosen clustering algorithm (e.g. K-medoids) with n^{child} clusters and metric d to obtain the clusters $\{S_j\}_j^{n^{child}}$
4:    evaluate the estimator of the error I for the refinement $\{S_j\}_j$, considering also the minimum and maximum reduced dimensions r_{min}, r_{max}
5:    **if** $I > b$ **and** the minimum number of elements n_{el} is not reached **and** the maximum number of clusters K is not reached **then**
6:       save the best refinement $\{S_j\}_j$ and update the best score b
7:    **end if**
8: **end for**

Remark 5 (Heuristics behind the choice of the active subspaces metric for K-medoids) Having in mind that the optimal profile $h(\mathbf{y}) = \mathbb{E}_{\mu_i}[f|P_{r(S_i),i}](\mathbf{y})$ from Definition 2 is approximated as $h(\mathbf{y}) = f(\mathbf{y}) = f(P_r\mathbf{x})$ as reported in Remark 2, we can argue that clustering with the AS metric from Eq. (15) is effective since, for this choice of the metric, the clusters tend to form transversally with respect to the active subspace directions. This is because the metric weights more the components with higher eigenvalues. So clustering with this metric reduces heuristically also the approximation error induced by the choice of the non-optimal profile.

Other clustering criterions employed must satisfy the subspace Poincaré inequality for each cluster. Regarding the regression method we employ Gaussian process regression with RBF-ARD kernel [43]. The procedure for response surface design with Gaussian processes and ridge approximation with active subspaces can be found in [1, 21]. As for the indicator for refinement (I), the local R^2 score in Remark 3 is employed to measure the accuracy of the ridge approximation against a validation dataset and the estimator from the RHS of Eq. 8 is used to determine the dimension of the active subspace of each cluster.

Here, we make some considerations about the complexity of the algorithm. For each refinement, considering an intermediate cluster of K elements, the most expensive tasks are: the active subspace evaluation $O((K/m)np^2 + (K/m)n^2p + n^3)$ (the first two costs refer to matrix multiplications, while the third for eigendecomposition), the clustering algorithm, for example K-medoids with AS distance $O(K(K-m)^2)$, and the Gaussian process regression $O((K/m)^3p^3)$, where p is the dimension of the outputs and $m = n_{min}^{child}$ and $M = n_{max}^{child}$ are the minimum and maximum number of children clusters, for a more compact notation. In the worst case the height of the refinement tree is $l = \log_m N/n_{el}$ where n_{el} is the minimum number of elements per cluster. In Sect. 6.4 we report the detailed computational costs associated to each refinement level.

4 Classification with Local Active Subspace Dimension

A poor design of the parameter space could add an avoidable complexity to the surrogate modeling algorithms. Often, in practical applications, each parameter range is chosen independently with respect to the others. Then, it is the responsibility of the surrogate modeling procedure to disentangle the correlations among the parameters. However, in this way, looking at the response surface from parameters to outputs, regions that present different degrees of correlation are treated indistinctly. In this matter, a good practice is to study as a preprocessing step some sensitivity measures, like the total Sobol' indices [33] among groups of parameters, and split the parameter space accordingly in order to avoid the use of more expensive surrogate modeling techniques later. Sobol' indices or the global active subspace sensitivity scores give summary statistics on the whole domain. So in general, one could study the parameter space more in detail, classifying nonlinearly regions with respect to the complexity of the response surface, if there are enough samples to perform such studies.

We introduce an effective approach to tackle the problem of classification of the parameter space with respect to a local active subspace information. With the latter we mean two possible alternatives.

Definition 3 (Local active subspace dimension) Given a threshold $\epsilon > 0$, the pairs of inputs and gradients $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_i$ associated to an objective function of interest $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, the size of the neighbourhood of sample points to consider $N \geq n$, and a subsampling parameter $p \in \mathbb{N}$, $p \leq N$, the local active subspace dimension r_i associated to a sample point $\mathbf{X}_i \in \mathcal{X}$ is the positive integer

$$r_i = \operatorname*{argmin}_{1 \leq r \leq p} \left\{ \operatorname{tr} \left((Id - P_r) \left(\frac{1}{p} \sum_{i \in J} \mathbf{dY}_i \otimes \mathbf{dY}_i \right) (Id - P_r) \right) \leq \epsilon \,\middle|\, J \in C(N, p) \right\},$$

where $C(N, p)$ is the set of combinations without repetition of the N elements of the Euclidean neighbourhood of $\mathbf{X}_i$ in p classes and P_r is the projection onto the first r eigenvectors of the symmetric positive define matrix

$$\frac{1}{p} \sum_{i \in J} \mathbf{dY}_i \otimes \mathbf{dY}_i.$$

Definition 4 (Local active subspace) Given the pairs of inputs and gradients $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_i$ associated to an objective function of interest $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, the size of the neighbourhood of sample points to consider $N \geq n$, and a fixed dimension $p \in \mathbb{N}$, $1 \leq p \leq N$, the local active subspace W_i associated to a sample point $\mathbf{X}_i \in \mathcal{X}$ is the matrix of the first p eigenvectors of the spectral decomposition of

$$\frac{1}{N} \sum_{i \in U} \mathbf{dY}_i \otimes \mathbf{dY}_i, \tag{16}$$

where U is the neighbourhood of sample points of $\mathbf{X}_i$ with respect to the Euclidean distance. In practice, we choose p close to the global active subspace dimension. The pairs $\{(\mathbf{X}_i, W_i)\}_i$ can be thought as a discrete vector bundle of rank p and $\{W_i\}_i$ can be thought as a subset of points of the Grassmannian $\mathrm{Gr}(N, p)$, that is the set of p-dimensional subspaces in an N-dimensional vector space.

Starting from the pairs of inputs-gradients $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_i$, the procedure follows these steps:

1. Each parameter sample is enriched with the additional feature corresponding to the local active subspace dimension from Definition 3 or the local active subspace from Definition 4, represented by the variable $\mathbf{Z}$.

2. Each sample $\mathbf{X}_i$ is labelled with an integer l_i that will be used as classification label in the next step. To label the pairs $\{(\mathbf{X}_i, \mathbf{Z}_i)\}_i$ we selected K-medoids with the Grassmannian metric

$$d((\mathbf{X}_i, \mathbf{Z}_i), (\mathbf{X}_j, \mathbf{Z}_j)) = \|\mathbf{Z}_i - \mathbf{Z}_j\|_F, \tag{17}$$

where $\|\cdot\|_F$ is the Frobenius distance, in case $\mathbf{Z}_i$ represents the local active subspace or spectral clustering [39] in case $\mathbf{Z}_i$ is the local active subspace dimension. In the last case, the labels correspond to the connected components of the graph built on the nodes $\{(\mathbf{X}_i, \mathbf{Z}_i)\}_i$ with adjacency list corresponding to the nearest nodes with respect to the distance

$$d((\mathbf{X}_i, \mathbf{Z}_i), (\mathbf{X}_j, \mathbf{Z}_j)) = \begin{cases} \infty, & \mathbf{Z}_i \neq \mathbf{Z}_j \\ \|\mathbf{X}_i - \mathbf{X}_j\|, & \mathbf{Z}_i = \mathbf{Z}_j \end{cases}, \tag{18}$$

where $\|\cdot\|$ is the Euclidean metric in $\mathbb{R}^n$. The connected components are obtained from the eigenvectors associated to the eigenvalue 0 of the discrete Laplacian of the graph [39]. Summarizing, we employ two labelling methods: K-medoids in case $\mathbf{Z}_i$ represents the local active subspace (Definition 4) W_i or spectral clustering in case $\mathbf{Z}_i$ represents the local active subspace dimension (Definition 3).

3. A classification method is applied to the inputs-labels pairs $\{(\mathbf{X}_i, l_i)\}_i$. Generally, for our relatively simple applications we apply a multilayer perceptron with 1000 hidden nodes and 2 layers.

Remark 6 (Grassmann distance) In general regarding the Definition 4, the dimension p could be varying among samples $\mathbf{X}_i$ and one could use a more general distance with respect to the one from Eq. (17) that can have as arguments two vectorial subspaces of possibly different and arbitrary large dimensions.

Remark 7 (Gradient-free active subspace) In general both the response surface design and the classification procedure above can be carried out from the pairs $\{(\mathbf{X}_i, \mathbf{Y}_i)\}_i$ of inputs, outputs instead of the sets $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_i$ of inputs, gradients. In fact, the gradients $\{\mathbf{dY}_i\}$ can be approximated in many different ways [1] from $\{(\mathbf{X}_i, \mathbf{Y}_i)\}_i$. In the numerical results in Sect. 5 when the gradients are not available they are approximated with the gradients of the local one-dimensional polynomial regression built on top of the neighbouring samples.

5 Numerical Results

In this section we apply the proposed localized AS method to some datasets of increasing complexity. We emphasize that the complexity is not only defined by the number of parameters but also by the intrinsic dimensionality of the problem. We compare the clustering techniques presented in Sect. 3, and we show how the active subspaces-based distance metric outperforms the Euclidean one for those functions which present a global lower intrinsic dimensionality. We remark that for hierarchical top–down clustering we can use both metrics, and we always show the best case for the specific dataset.

We start from a bidimensional example for which we can plot the clusters and the regressions, and compare the different techniques. Even if it is not a case for which one should use

---

Algorithm 4 Classification with local features from the AS information.

Input: inputs-gradients pairs $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_{i \in I}$ as training dataset
local features based on AS information $\{\mathbf{Z}_i\}_{i \in I}$
labelling method based on the distance d from Eqs. (17) or (18)
classification method taking the inputs-labels pairs $\{(\mathbf{X}_i, l_i)\}_{i \in I}$ **Output**: predictor for new
test inputs and classes of the training dataset.

1: **for each** $i \in I$ **do**
2:     evaluate feature $\mathbf{Z}_i$ from $(\mathbf{X}_i, \mathbf{dY}_i)$ and the neighbouring points of $\mathbf{X}_i$
3: **end for**
4: initialize the $|I| \times |I|$ distance matrix M associated to $\{(\mathbf{X}_i, \mathbf{Z}_i)\}_{i \in I}$
5: **for each** $i \in I$ **do**
6:     **for each** $i \leq j \in I$ **do**
7:         $M(i, j) = d((\mathbf{X}_i, \mathbf{Z}_i), (\mathbf{X}_j, \mathbf{Z}_j))$
8:     **end for**
9: **end for**
10: use the labelling method with input M, to assign a label l_i for each $(\mathbf{X}_i, \mathbf{Z}_i)$
11: train the classification method with the inputs-labels pairs $\{(\mathbf{X}_i, l_i)\}_{i \in I}$

---

parameter space dimensionality reduction we think it could be very useful for the reader to understand also visually all the proposed techniques. For the higher dimensional examples we compare the accuracy of the methods in terms of R^2 score and classification performance. All the computations regarding AS are done with the open source Python package[1] called ATHENA [44], for the classification algorithms we use the scikit-learn package [45], and for the Gaussian process regression GPy [46].

We suppose the domain $\mathcal{X}$ to be a n-dimensional hyperrectangle. we are going to rescale the input parameters $\mathbf{X}$ to $[-1, 1]^n$.

5.1 Some Illustrative Bidimensional Examples

We start by presenting two bidimensional test cases to show every aspect of the methodology together with illustrative plots. First we analyse a case where a global active subspace, even if present, does not provide a regression accurate enough along the active direction, in Sect. 5.1.1. Then we consider a radial symmetric function for which, by construction, an AS does not exist, in Sect. 5.1.2, and the use of K-means is instead preferable since we cannot exploit a privileged direction in the input domain.

5.1.1 Quartic Function

Let us consider the following bidimensional quartic function $f(\mathbf{x}) = x_1^4 - x_2^4$, with $\mathbf{x} = (x_1, x_2) \in [0, 1]^2$. In Fig. 1 we can see the contour plot of the function, the active subspace direction — translated for illustrative reasons — and the corresponding sufficient summary plot of the global active subspace, computed using 400 uniformly distributed samples. With sufficient summary plot we intend $f(\mathbf{x})$ plotted against the input parameters projected onto the active subspace, that is $W_1^T \mathbf{x}$. It is clear how, in this case, a univariate regression does not produce any useful prediction capability.

Let us apply the clustering techniques introduced in the previous sections fixing the number of clusters to 4. In Fig. 2 we can clearly see how the supervised distance metric in Eq. (15) acts in dividing the input parameters. On the left panel we apply K-means which clusters

---

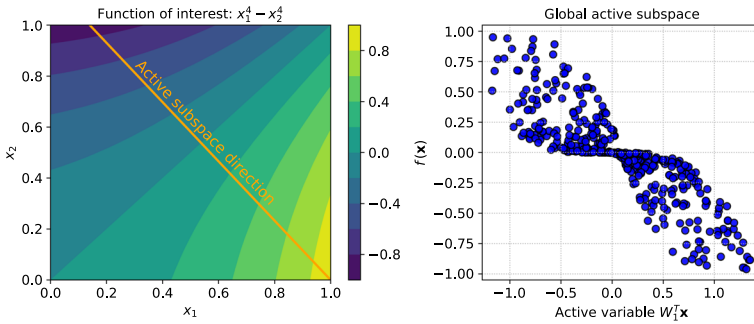[1] Available at https://github.com/mathLab/ATHENA/.

Fig. 1 On the left panel the contour plot of the quartic function and in orange the global active subspace direction. On the right panel the sufficient summary plot resulting projecting the data onto the global AS
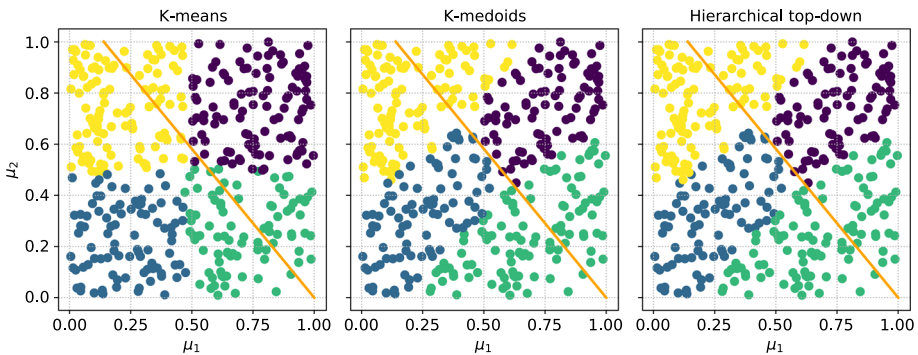

Fig. 2 Comparison between the different clusters obtained by K-means (on the left), K-medoids (middle panel), and hierarchical top–down (on the right) with AS induced distance metric defined in Eq. (15) for the quartic test function. In orange the global active subspace direction. Every cluster is depicted in a different color (color figure online)

the data into 4 uniform quadrants, while in the middle and right panels we have K-medoids and hierarchical top–down, respectively, with a subdivision aligned with the global AS. We notice that for this simple case the new metric induces an identical clustering of the data. In Fig. 3 we plotted the sufficient summary plots for each of the clusters individuated by K-medoids or hierarchical top–down in Fig. 2. By using a single univariate regression for each cluster the R^2 score improves a lot with respect to a global approach (see right panel of Fig. 1).

We can also compare the R^2 scores for all the methods, using a test datasets of 600 samples. In Fig. 4 we report the scores for K-means, K-medoids and for hierarchical top–down with AS-based distance metric. The score for the global AS, which is 0.78, is not reported in Fig. 4 for illustrative reasons. The results are very similar due to the relatively simple test case, but we can see that even with 2 clusters the gain in accuracy is around 23% using the metric in Eq. (15).

The hierarchical top–down clustering method was ran with the following hyper-parameters: the total number of clusters is increasing from 2 to 10, the minimum number of children equal to the maximum number of children equal to 3, uniform normalization of the clusters, the minimum size of each cluster is 10 elements, the clustering method is K-medoids with AS distance, the maximum active subspace dimension is 1.
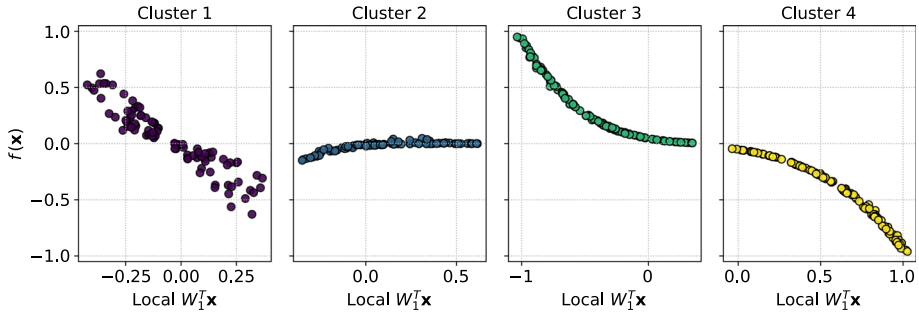
Fig. 3 Local sufficient summary plots for the 4 clusters individuated by K-medoids or hierarchical top–down in Fig. 2 (colors correspond) (color figure online)
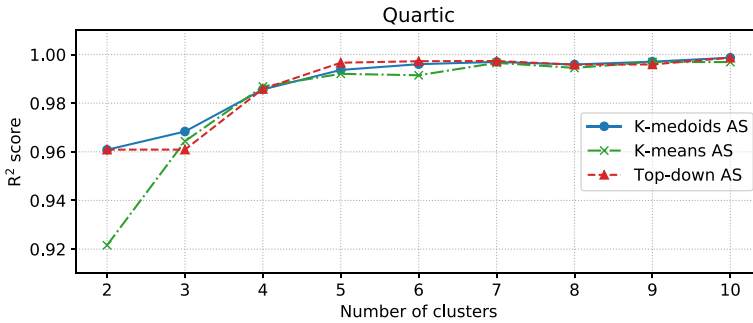

Fig. 4 R^2 scores comparison between local versions varying the number of clusters for the quartic function. Global AS has a score equal to 0.78
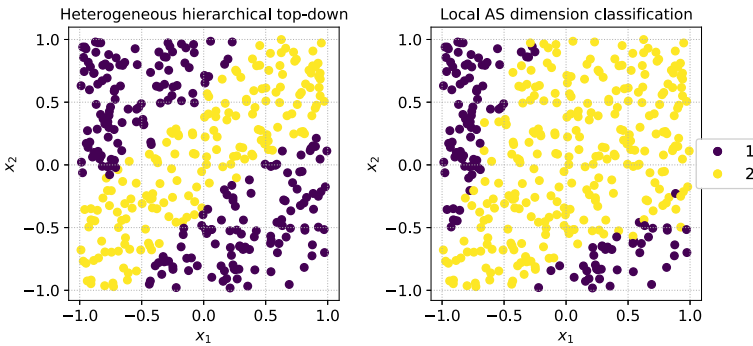

Fig. 5 On the left panel the hierarchical top–down clustering with heterogeneous AS dimension and R^2 score equal to 1. On the right panel the labels of the local AS dimension from Definition 3

Then we want to increase the accuracy of the regression for a fixed number of clusters equal to 3, loosing in some regions the reduction in the parameter space. Starting from the clustering with hierarchical top–down and 3 clusters of dimension 1, the AS dimension of each of the 3 clusters is increased if the threshold of 0.95 on the local R^2 score is not met. In general, the local R^2 score is evaluated on a validation set, for which predictions from the local response surfaces are obtained, after each validation sample is classified into one of the 3 clusters.
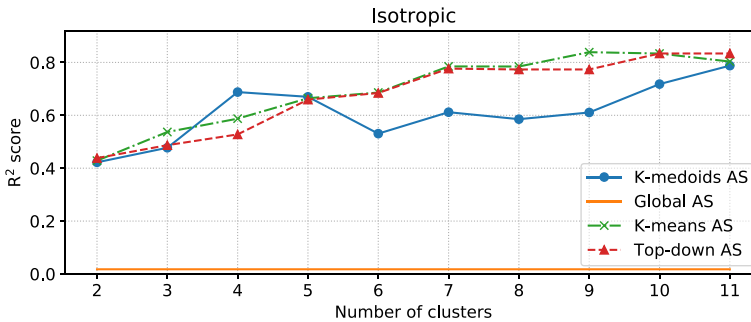
Fig. 6 R^2 scores comparison between global AS and local versions varying the number of clusters for the isotropic model function. Global AS corresponds to no clustering

The 3 clusters are reported in Fig. 5 on the left. The R^2 score on the test set is 1, instead of around 0.97 from Fig. 4. To obtain this result, the central cluster AS dimension is increased from 1 to 2. We compare the clustering with respect to the classification of the local AS dimension with algorithm 4 using as features the local AS dimension as defined in Definition 3, on the right of Fig. 5. Actually, algorithm 4 is stopped after the plotted labels are obtained as the connected components of the underlying graph to which spectral clustering is applied: no classification method is employed, yet. It can be seen that hierarchical top–down clustering with heterogeneous AS dimension is more efficient with respect to the classes of algorithm 4, regarding the number of samples associated to a response surface of dimension 2.

5.1.2 Radial Symmetric Cosine

This example addresses the case for which an active subspace is not present. This is due to the fact that there are no preferred directions in the input domain since the function f has a radial symmetry. For this case the exploitation of the supervised distance metric does not provide any significant gain and K-means clustering works better on average, since it does not use the global AS structure. The model function we consider is $f(\mathbf{x}) = \cos(\|\mathbf{x}\|^2)$, with $\mathbf{x} \in [-3, 3]^2$.

In Fig. 6 we compare the R^2 scores for K-means, K-medoids with AS-based metric, and hierarchical top–down with Euclidean metric. We used 500 training samples and 500 test samples. We see K-medoids has not a clear behaviour with respect to the number of clusters, while the other methods present a monotonic trend and better results on average, especially K-means. On the other hand local models improve the accuracy considerably, even for a small number of clusters, with respect to a global model.

In this case the specifics of hierarchical top–down clustering are: the minimum number of children is equal to the maximum, the minimum number of elements per cluster is 10, the clustering method chosen is K-means, the normalization employed it the uniform one, and the total number of clusters is increasing from 2 to 11.

5.2 Higher-Dimensional Datasets

In this section we consider more interesting benchmarks, for which dimension reduction in the parameter space is useful since the starting dimension of the parameter space is higher. We test
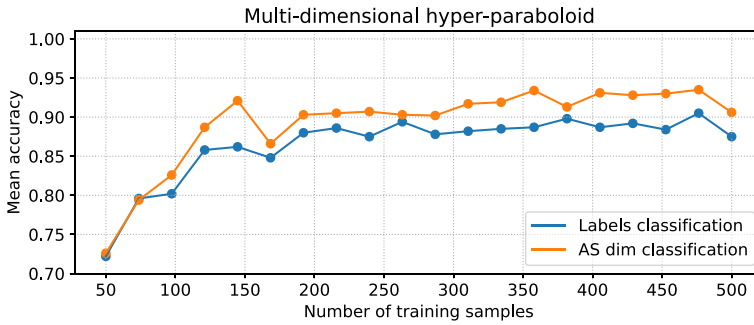
Fig. 7 Mean accuracy study for a training dataset increasing in size from 50 to 500 samples. The test set is made of 1000 independent samples. The classification accuracy for the procedures of connected component classification (in blue) and local AS dimension classification (in orange) are both shown (color figure online)

the classification procedure in algorithm 4 with an objective function with 6 parameters and defined piecewise as a paraboloid with different AS dimensions. We also test the procedure of response surface design with local AS, with a classical 8-dimensional epidemic benchmark model.

5.2.1 Multi-dimensional Hyper-Paraboloid

The objective function $f : [-4, 4]^6 \to \mathbb{R}$ we consider is defined piecewise as follows

$$f(x) = \begin{cases} x_1^2 & \text{if } x_1 > 0 \text{ and } x_2 > 0, \\ x_1^2 + x_2^2 & \text{if } x_1 < 0 \text{ and } x_2 > 0, \\ x_1^2 + x_2^2 + x_3^2 & \text{if } x_1 > 0 \text{ and } x_2 < 0, \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 & \text{if } x_1 < 0 \text{ and } x_2 < 0. \end{cases} \tag{19}$$

In the 4 domains in which f is defined differently, we expect an AS dimension ranging from 1 to 4, respectively. We employed Algorithm 4 using the local AS dimensions as additional features, from Definition 3: the values of the hyper-parameters are the following: $\epsilon = 0.999$, $N = 6$, $p = 4$. In Fig. 7 we plot the accuracy of the classification of the labels, associated to the connected components of the graph built as described in Algorithm 4, and also the accuracy of the classification of the local active subspace dimension, that takes the values from 1 to 4. The test dataset for both the classification errors has size 1000. The score chosen to asses the quality of the classification is the mean accuracy, that is the number of correctly predicted labels over the total number of labels. For both the classification tasks 100 train samples are enough to achieve a mean accuracy above 80%.

We remark that every step is applied to a dataset of samples in a parameter space of dimension 6, even if, to get a qualitative idea of the performances of the method, in Fig. 8 we show only the first two components of the decision boundaries of the 4 classes for both the previously described classification problems.

5.2.2 Ebola Epidemic Model

In this section we examine the performance of the proposed methods over the dataset created with the SEIR model for the spread of Ebola.[2] The output of interest in this case is the basic

---

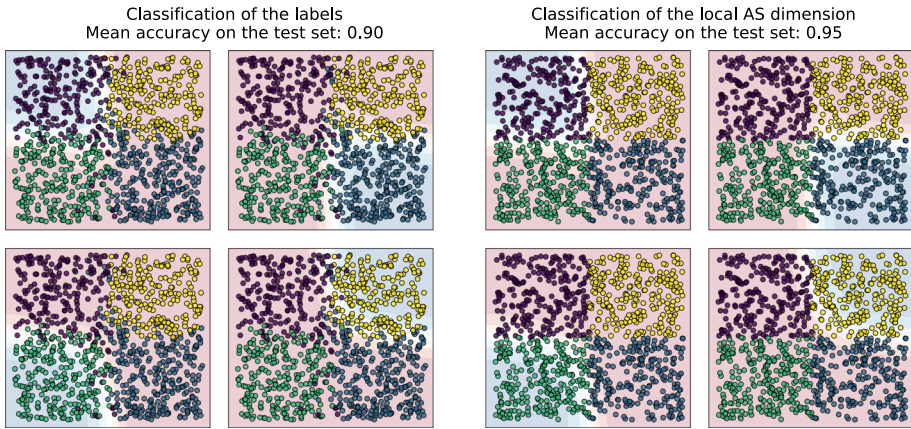[2] The dataset was taken from https://github.com/paulcon/as-data-sets.

Fig. 8 On the left panel, the decision boundaries of the 4 classes associated to the connected components of the graph built as described in Algorithm 4. On the right panel, the decision boundaries of the 4 classes associated to the local AS dimension from 1 to 4. The datasets has dimension 6, only the first two components of the decision boundaries and of the test samples are plotted

reproduction number R_0 of the SEIR model, described in [47], which is computed using 8 parameters as follows

$$R_0 = \frac{\beta_1 + \frac{\beta_2 \rho_1 \gamma_1}{\omega} + \frac{\beta_3}{\gamma_2}\psi}{\gamma_1 + \psi}. \tag{20}$$

As shown in previous works, this function has a lower intrinsic dimensionality, and thus a meaningful active subspace, in particular of dimension 1. To evaluate the performance of the local AS we compute the R^2 score, as in Eq. (11), varying the number of clusters from 2 to 10 for all the methods presented. The test and training datasets are composed by 500 and 300, respectively, uniformly distributed and independent samples. The results are reported in Fig. 9, where as baseline we reported the R^2 for the GPR over the global AS. We can see how the use of the AS-based distance metric contributes the most with respect to the actual clustering method (compare K-medoids and hierarchical top–down in the plot). K-means, instead, does not guarantee an improved accuracy (for 4 and 9 clusters), and in general the gain is limited with respect to the other methods, especially for a small number of clusters which is the most common case in practice, since usually we work in a data scarcity regime. The results for K-medoids and top–down are remarkable even for a small amount of clusters with an R^2 above 0.9 and an improvement over 10% with respect to the global AS, which means that no clustering has been used.

The hyper-parameters for the hierarchical top–down algorithm are the following: the maximum local active subspace dimension is 1, the maximum number of children is equal to the number of total clusters, the minimum number of children is 2 at each refinement level, the minimum number of elements per cluster is 10, and the clustering method for each refinement is K-medoids with AS distance.

5.3 Datasets with Vectorial Outputs

In this section we want to show how hierarchical top–down clustering and the classification procedure of algorithm 4 can be combined to improve the overall reduction in the parameter
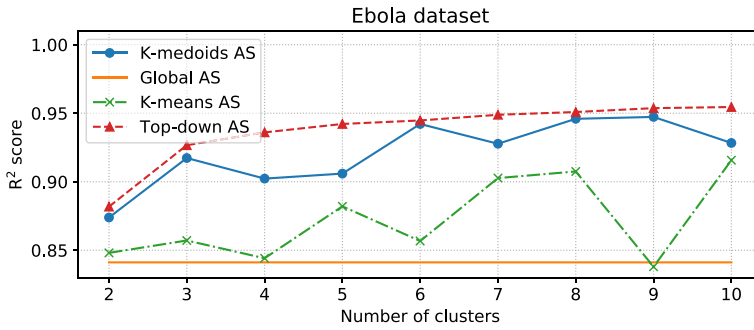
Fig. 9 R^2 scores comparison between global AS and local versions varying the number of clusters for the Ebola spread model. Global AS corresponds to no clustering

space, for a fixed lower threshold in the R^2 score. For the response surface design with active subspaces for vectorial outputs we refer to [21, 34].

5.3.1 Poisson Equation with Random Diffusivity

Let us consider the stochastic Poisson problem on the square $\mathbf{x} = (x, y) \in \Omega := [0, 1]^2$, defined as:

$$
\begin{cases}
-\nabla \cdot (\kappa \, \nabla u) = 1, & \mathbf{x} \in \Omega, \\
u = 0, & \mathbf{x} \in \partial\Omega_{\text{top}} \cup \partial\Omega_{\text{bottom}}, \\
u = 10y(1 - y), & \mathbf{x} \in \partial\Omega_{\text{left}}, \\
\mathbf{n} \cdot \nabla u = 0, & \mathbf{x} \in \partial\Omega_{\text{right}},
\end{cases} \tag{21}
$$

with homogeneous Neumann boundary condition on $\partial\Omega_{\text{right}}$, and Dirichlet boundary conditions on the remaining part of $\partial\Omega$. The diffusion coefficient $\kappa : (\Omega, \mathcal{A}, P) \times \Omega \to \mathbb{R}$, with $\mathcal{A}$ denoting a σ-algebra, is such that $\log(\kappa)$ is a Gaussian random field, with covariance function $G(\mathbf{x}, \mathbf{y})$ defined by

$$
G(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{\beta^2}\right), \quad \forall \mathbf{x}, \mathbf{y} \in \Omega, \tag{22}
$$

where the correlation length is $\beta = 0.03$. We approximate this random field with the truncated Karhunen-Loève decomposition as

$$
\kappa(s, \mathbf{x}) \approx \exp\left(\sum_{i=1}^{m} X_i(s)\gamma_i \boldsymbol{\psi}_i(\mathbf{x})\right), \quad \forall (s, \mathbf{x}) \in \Omega \times \Omega, \tag{23}
$$

where $(X_i)_{i \in 1,\dots,m}$ are independent standard normal distributed random variables, and the eigenpairs of the Karhunen-Loève decomposition of the zero-mean random field κ are denoted with $(\gamma_i, \boldsymbol{\psi}_i)_{i \in 1,\dots,m}$. The parameters $(X_i)_{i \in 1,\dots,m=10}$ sampled from a standard normal distribution are the coefficients of the Karhunen-Loève expansion, truncated at the first 10 modes, so the parameter space has dimension $m = 10$.

The domain Ω is discretized with a triangular unstructured mesh $\mathcal{T}$ with 3194 triangles. The simulations are carried out with the finite element method with polynomial order 1. The solution u is evaluated at 1668 degrees of freedom, thus the output is vectorial with dimension $d = 1668$. As done in [21, 34], the output is enriched with the metric induced by
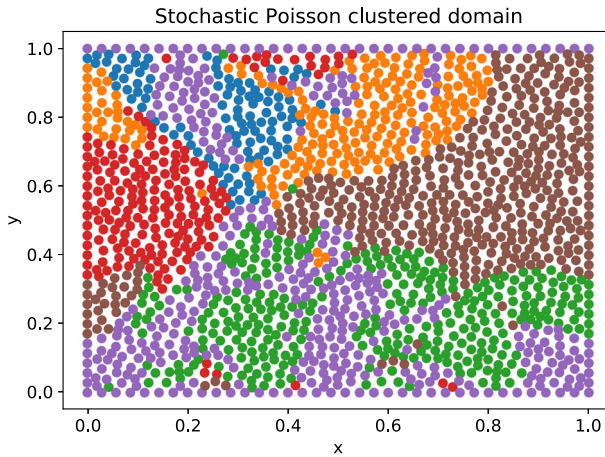
Fig. 10 Subdivision of the spatial domain Ω in 6 clusters based on the Grassmann distance from Definition 4, i.e. the clusters correspond to the connected components of the graph built on top of the degrees of freedom with adjacency list determined using as distance Definition 4

the Sobolev space $H^1(\Omega)$ on to the finite element space of polynomial order 1: the metric is thus represented by a $d \times d$ matrix M obtained as the sum of the mass and stiffness matrices of the numerical scheme and it is involved in the AS procedure when computing the correlation matrix $\mathbb{E}\left[Df \, M \, Df^T\right]$, where Df is the $m \times d$ Jacobian matrix of the objective function $f : \mathbb{R}^{10} \to \mathbb{R}^d$, that maps the first $m = 10$ coefficients of the Karhunen-Loéve expansion $(X_i)_{i \in 1,\dots,m}$ to the solution u. The Jacobian matrix is evaluated for each set of parametric instances with the adjoint method, as in [21].

Since the output is high-dimensional we classified with Algorithm 4 the output space in 6 clusters, using the Grassmann distance from Eq. (17), as shown in Fig. 10.

Afterwards we applied hierarchical top–down clustering to every one of the 6 triplets of inputs-outputs-gradients, obtained restricting the outputs and the gradients to each one of the 6 clusters. The specifics of hierarchical top–down clustering we employed are the following: the minimum and maximum number of children for each refinement are equal to the total number of clusters, which is 4, the minimum number of elements in each cluster is 10, and the clustering algorithm chosen is K-medoids with the AS distance. The size of the training and test datasets is respectively of 500 and 150. The gradients are evaluated with the adjoint method. Since the output is vectorial we employed the mean R^2 score, where the average is made among the components of the vectorial output considered.

Then for every lower threshold on the R^2 score we increase one by one the dimension of the 6×4 local clusters, until all the R^2 scores of each of the 6 triplets are above the fixed threshold. The same procedure is applied to the whole dataset of inputs-outputs-gradients but executing hierarchical top–down clustering just once, for all the output's components altogether.

The results are reported in Fig. 11. In the case of the clustered outputs, the local dimension of each one of the 6 clustered outputs times 4 local clusters in the parameter space, for a total of 24 local clusters, are weighted with the number of elements of each cluster. In the same way the 4 clusters of the case with unclustered outputs is weighted with the number of the elements of each one of the 4 clusters. It can be seen that for every fixed threshold, there is an evident gain, with respect to the dimension reduction in the parameter space, in clustering the outputs and then performing hierarchical top–down clustering in the parameter space.
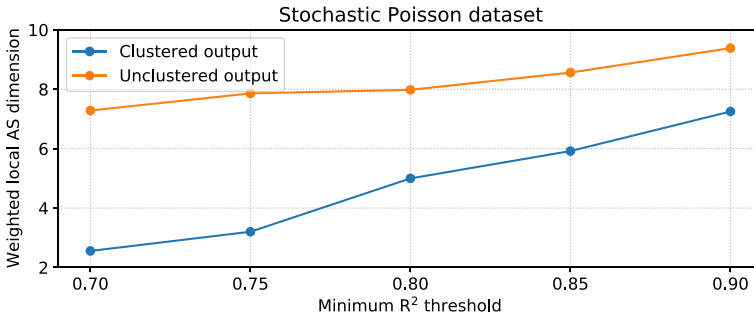
Fig. 11 In orange the local AS dimensions weighted on the number of elements of each of the 4 clusters in the parameter space, obtained with hierarchical top–down clustering. In blue the local AS dimensions weighted on the number of elements of each of the 4 clusters in the parameter space, obtained with hierarchical top–down clustering, times 6 clustered outputs (see Fig. 10) for a total of 24 terms in the weighted average (color figure online)

5.3.2 Shape Design of an Airfoil

For this vectorial test case we consider the temporal evolution of the lift coefficient of a parametrized NACA airfoil. Here we briefly present the problem we solve to create the dataset, we refer to [48] for a deeper description.

Let us consider the unsteady incompressible Navier–Stokes equations described in an Eulerian framework on a parametrized space-time domain $S(\boldsymbol{\mu}) = \Omega(\boldsymbol{\mu}) \times [0, T] \subset \mathbb{R}^2 \times \mathbb{R}^+$. The vectorial velocity field $\mathbf{u} : S(\boldsymbol{\mu}) \to \mathbb{R}^2$, and the scalar pressure field $p : S(\boldsymbol{\mu}) \to \mathbb{R}$ solve the following parametric PDE:

$$\begin{cases} \mathbf{u_t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot 2\nu \nabla^s \mathbf{u} = -\nabla p & \text{in } S(\boldsymbol{\mu}), \\ \nabla \cdot \mathbf{u} = \mathbf{0} & \text{in } S(\boldsymbol{\mu}), \\ \mathbf{u}(t, \mathbf{x}) = \mathbf{f}(\mathbf{x}) & \text{on } \Gamma_{\text{in}} \times [0, T], \\ \mathbf{u}(t, \mathbf{x}) = \mathbf{0} & \text{on } \Gamma_0(\boldsymbol{\mu}) \times [0, T], \\ (\nu \nabla \mathbf{u} - p\mathbf{I})\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{\text{out}} \times [0, T], \\ \mathbf{u}(0, \mathbf{x}) = \mathbf{k}(\mathbf{x}) & \text{in } S(\boldsymbol{\mu})_0 \end{cases} \tag{24}$$

Here, $\Gamma = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_0$ denotes the boundary of $\Omega(\boldsymbol{\mu})$ composed by inlet boundary, outlet boundary, and physical walls, respectively. With $\mathbf{f}(\mathbf{x})$ we indicate the stationary non-homogeneous boundary condition, and with $\mathbf{k}(\mathbf{x})$ the initial condition for the velocity at $t = 0$. The geometrical deformation are applied to the boundary $\Gamma_0(\boldsymbol{\mu})$. The undeformed configuration corresponds to the NACA 4412 wing profile [49, 50]. To alter such geometry, we adopt the shape parametrization and morphing technique proposed in [51], where 5 shape functions are added to the airfoil profiles. They are commonly called Hicks-Henne bump functions. Let y_u and y_l be the upper and lower ordinates of the profile, respectively. The deformation of such coordinates is described as follows

$$y_u = \overline{y_u} + \sum_{i=1}^{5} c_i r_i, \qquad y_l = \overline{y_l} - \sum_{i=1}^{5} d_i r_i, \tag{25}$$

where the bar denotes the reference undeformed state. The parameters $\boldsymbol{\mu} \in \mathbb{D} \subset \mathbb{R}^{10}$ are the weights coefficients, c_i and d_i, associated with the shape functions r_i. In particular we set $\mathbb{D} := [0, 0.03]^{10}$. The explicit formulation of the shape functions can be found in [51]. For
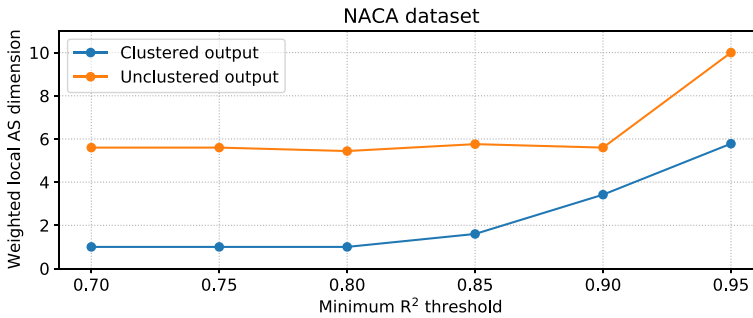
Fig. 12 In orange the local AS dimensions weighted on the number of elements of each of the 2 clusters in the parameter space, obtained with hierarchical top–down clustering. In blue the local AS dimensions weighted on the number of elements of each of the 2 clusters in the parameter space, obtained with hierarchical top–down clustering, times 2 clustered outputs for a total of 4 terms in the weighted average (color figure online)

this datasets, the Reynolds number is $Re = 50000$. The time step is $dt = 10^{-3}$ s. For other specifics regarding the solver employed and the numerical method adopted we refer to [48].

As outputs we considered the values of the lift coefficient, every 15 time steps from 100 ms to 30000 ms, for a total of 1994 components. Even in this case the output is classified with Algorithm 4 with distance defined in Definition 3. The values of the lift coefficient physically interesting are collected at last, after an initialization phase. Nonetheless for the purpose of having a vectorial output we considered its value from the time instant 100 ms. The procedure finds two classes and splits the ordered output components in two parts: from the component 0 to 996, the local AS dimension is 1, for the remaining time steps it is higher. So we can expect an improvement on the efficiency of the reduction in the parameter space when considering separately these two sets of outputs components as Fig. 12 shows. The weighted local AS dimension is in fact lower when using clustering, for every minimum R^2 threshold.

6 Conclusions and Perspectives

In this work we present a new local approach for parameter space reduction which exploits supervised clustering techniques, such as K-means, K-medoids, and hierarchical top–down, with a distance metric based on active subspaces. We called this method local active subspaces (LAS). The proposed metric tend to form the clusters transversally with respect to the active subspace directions thus reducing the approximation error induced by the choice of the non-optimal profile.

The theoretical formulation provides error estimates for the construction of response surfaces over the local active subspaces. We also present a classification approach to capture the optimal AS dimension for each cluster and can be used as a preprocessing step, both for the inputs and the vectorial outputs, for the construction of more accurate regressions and surrogate modeling. The proposed approach is very versatile, especially the hierarchical top–down clustering which can incorporate quite different criteria. The methodology has been validated over a vast range of datasets, both scalar and vector-valued, showing all the strengths and a possible weakness, in case of radial symmetric functions. In all the test cases LAS achieved superior performance with respect to the classical global approach.

Possible future lines of research can focus on the study of the extension of this methods to nonlinear parameter space reduction techniques, or on the use of more advanced clustering criteria.

Appendix

6.1 Subspace Poincaré Inequality

The probabilistic Poincaré inequality for conditional probability densities or subspace Poincaré inequality [36] is valid at least for the following classes of absolutely continuous probability densities μ with p.d.f. ρ.

Assumption 1 The p.d.f $\rho : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ satisfies one of the following:

1. $\mathcal{X}$ is bounded connected open with Lipschitz boundary, ρ is the uniform density distribution.
2. $\mathcal{X}$ is convex and bounded, $\exists\, \delta, D > 0 : 0 < \delta \leq \|\rho(\mathbf{x})\|_{L^\infty} \leq D < \infty\ \forall \mathbf{x} \in \mathcal{X}$,
3. $\mathcal{X} = \mathbb{R}^n$, $\rho(\mathbf{x}) \sim \exp(-V(\mathbf{x}))$ where $V : \mathbb{R}^n \to (-\infty, \infty]$, $V \in \mathcal{C}^2$ is α-uniformly convex,

$$\mathbf{u}^T \mathrm{Hess}(V(\mathbf{x}))\mathbf{u} \geq \alpha \|\mathbf{u}\|_2^2, \quad \forall \mathbf{x}, \mathbf{u} \in \mathbb{R}^n, \tag{26}$$

   where $\mathrm{Hess}(V(\mathbf{x}))$ is the Hessian of $V(\mathbf{x})$.
4. $\mathcal{X} = \mathbb{R}^n$, $\rho(\mathbf{x}) \sim \exp(-V(\mathbf{x}))$ where V is a convex function. In this case we require also f Lipschitz continuous.

The last class of p.d.f. provides a weaker bound (Lemma 4.3, [36]) on the ridge approximation error. For the previous classes $i \in \{1, 2, 3, 4\}$ of p.d.f. an upper bound of the Poincaré constant $C_{P,i}$ is also provided:

$$C_{P,1} = C_{P,1}(\Omega), \qquad C_{P,2} = \frac{D\,\mathrm{diam}(\mathcal{X})}{\pi \delta}, \qquad C_{P,3} = \frac{1}{\alpha}, \tag{27}$$

while the upper bound for $C_{P,4}$ requires the definition of other quantities and is proved in Lemma 4.4 [36].

6.2 Generalization of the Upper Bound on the Approximation of the Active Subspace

We want to make some brief considerations about the accuracy of the active subspace as eigen subspace of the correlation matrix approximated with Monte Carlo. If we use the notation $W_1 \in \mathbb{R}^{n \times r}$, $W_2 \in \mathbb{R}^{n \times (n-r)}$ for the active and inactive subspaces (i.e. $P_r = W_1 W_1^T$, $Id - P_r = W_2 W_2^T$) and $\hat{W}_1 \in \mathbb{R}^{n \times r}$, $\hat{W}_2 \in \mathbb{R}^{n \times (n-r)}$ for the approximated active and inactive subspaces, we can bound the approximation error as done by Constantine in [1]: assuming f Lipschitz continuous, with high probability the following inequality is valid,

$$\mathrm{dist}(\mathrm{Im}(W_1), \mathrm{Im}(\hat{W}_1)) \lesssim \frac{4L\sqrt{n}(\log(n))^{\frac{1}{2}}}{N^{\frac{1}{2}} \lambda_1 (\lambda_r - \lambda_{r+1})}, \tag{28}$$

where L is the Lipschitz constant of f, $\{\lambda_1, \dots \lambda_n\}$ are the non-negative eigenvalues of $\mathbb{E}_{\mu_i}[\nabla f \otimes \nabla f]$ ordered decreasingly, and N is the number of Monte Carlo samples.

The bound in Eq. (28) is obtained from Corollaries 3.8 and 3.10 in [1]. It is founded on a matrix Bernstein inequality for a sequence of random uniformly bounded matrices (Theorem 6.1, [52]) and on the Corollary 8.1.11 from [53] that holds a bound on the sensitivity of perturbation of an invariant subspace.

From Corollary 8.1.11 of [53], a bound on the approximation error of the active subspace W_1 can be obtained making explicit $\|W_2^T E W_1\|_F$ with respect to the chosen numerical method for the discretization $\hat{C}$ of the integral $C = \mathbb{E}_{\mu_i}[\nabla f \otimes \nabla f]$: in [1] this has been done for the Monte Carlo method. In practice we could use quasi Monte Carlo sampling methods with Halton or Sobol' sequences [33], since

$$
\begin{aligned}
\|W_2^T E W_1\|_F &\leq \sqrt{r(n-r)}\|W_2^T E W_1\|_{\max} \\
&\lesssim \sqrt{r(n-r)}D^*(\{x_i\}_i) \cdot \max_{i,j \in \{1,\dots,n\}}(V^{\mathrm{HK}}(\nabla f_i \nabla f_j)) \\
&\lesssim 2\sqrt{r(n-r)}D^*(\{x_i\}_i) \cdot \max(|f|) \cdot \max_{i \in \{1,\dots,n\}}(V^{\mathrm{HK}}(\nabla f_i)) \\
&\lesssim 2\sqrt{r(n-r)} \cdot \max_{i \in \{1,\dots,n\}}(V^{\mathrm{HK}}(\nabla f_i))\frac{\log(N)^n}{N},
\end{aligned}
$$

where V^{HK} is the Hardy–Krause variation and $D^*(\{x_i\}_i)$ is the star discrepancy of the quasi random sequence $\{x_i\}_i$. For the above result we have imposed $\mathcal{X} = [0,1]^n$ but it can be extended to different domains [54]. Thus we obtain the bound

$$
\begin{aligned}
\mathrm{dist}(\mathrm{Im}(W_1), \mathrm{Im}(\hat{W}_1)) &\lesssim \frac{4\|W_2^T E W_1\|_F}{\lambda_r - \lambda_{r+1}} \\
&\lesssim \frac{8L\sqrt{r(n-r)} \cdot \max_{i \in \{1,\dots,n\}}(V^{\mathrm{HK}}(\nabla f_i))}{\lambda_r - \lambda_{r+1}} \cdot \frac{\log(N)^n}{N}. \quad (29)
\end{aligned}
$$

Other numerical integration rules can be chosen so that different regularity conditions on the objective function may appear on the upper bound of the error, as the Lipschitz constant on Eq. (28) or the Hardy–Krause variation on Eq. (29). If the regularity of f is $\mathcal{C}^s$, we can also apply tensor product quadrature formulae or Smolyak's sparse quadrature rule [33]. For high-dimensional datasets and f less regular, the estimate in Eq. (28) is the sharpest.

6.3 Proof of Corollary 1

Proof Let us use the notation $h_1(x_1) := x_1(x_1 + \epsilon)(x_1 - \epsilon)$, and $h_2(x_2) := \cos(\omega x_2)$, it can be shown that

$$
\begin{aligned}
\mathbb{E}_\mu[\nabla f \otimes \nabla f] &= \int_B \begin{pmatrix} (h_1')^2(h_2)^2 & h_1 h_1' h_2 h_2' \\ h_1 h_1' h_2 h_2' & (h_1)^2(h_2')^2 \end{pmatrix} d\mu(\mathbf{x}) + \mu(A \cup C) \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} \frac{2}{5}\epsilon^5\left(1 + \frac{\sin(2\omega)}{2\omega}\right) & 0 \\ 0 & \frac{4}{105}\omega^2\epsilon^7\left(1 - \frac{\cos(2\omega)}{2\omega}\right) \end{pmatrix} + \mu(A \cup C) \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},
\end{aligned}
$$

thus, since we are considering a one dimensional active subspace, the active eigenvector belongs to the set $\{(1,0),(0,1)\}$. Similarly we evaluate

$$
\mathbb{E}_{\mu_B}[\nabla f|_B \otimes \nabla f|_B] = \begin{pmatrix} \frac{8}{5}\epsilon^4\left(1 + \frac{\sin(2\omega)}{2\omega}\right) & 0 \\ 0 & \frac{16}{105}\omega^2\epsilon^6\left(1 - \frac{\cos(2\omega)}{2\omega}\right) \end{pmatrix},
$$

$$
\mathbb{E}_{\mu_A}[\nabla f|_A \otimes \nabla f|_A] = \mathbb{E}_{\mu_C}[\nabla f|_C \otimes \nabla f|_C] = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},
$$

and conclude that there exist $\epsilon > 0$, $\omega > 0$ such that:

Table 1 Computational complexity of hierarchical top–down clustering

Step	Cost	Description
Root	$O(Nnp^2 + Nn^2p + n^3)$	AS
	$O(N^3p^3)$	GPR
First refinement	$O(N(N-k)^2)$	K-medoids
k from m to M	$O((N/k)np^2 + (N/k)n^2p + n^3)$	AS
	$O((N/k)^3p^3)$	GPR
Intermediate refinements	–	–
Last refinement	$O((N/k^{l-1})((N/k^{l-1})-k)^2)$	K-medoids
k from m to M	$O((N/k^l)np^2 + (N/k^l)n^2p + n^3)$	AS
for each one of the m^{l-1} clusters	$O((N/k^l)^3p^3)$	GPR

$$\frac{2}{5}\epsilon^5\left(1+\frac{\sin(2\omega)}{2\omega}\right)+4(1-\epsilon)\geq\frac{4}{105}\omega^2\epsilon^7\left(1-\frac{\cos(2\omega)}{2\omega}\right), \tag{30}$$

$$\frac{8}{5}\epsilon^4\left(1+\frac{\sin(2\omega)}{2\omega}\right)\leq\frac{16}{105}\omega^2\epsilon^6\left(1-\frac{\cos(2\omega)}{2\omega}\right), \tag{31}$$

for example $\epsilon \sim 10^{-2}$, $\omega \sim 10^4$ (approximately $10\epsilon^{-2} \leq \omega^2 \leq 10\epsilon^{-7}$). In this way, using the notations of Definition 2, we have

$$P_{1,\mathcal{X}} = e_1 \otimes e_1, \quad P_{1,A} = P_{1,C} = e_1 \otimes e_1, \quad P_{1,B} = e_2 \otimes e_2,$$

and it follows that

$$\mathbb{E}_\mu\left[\|f - R_{AS}(r,f)\|^2\right] = \mathbb{E}_\mu\left[f^2|_B\right] = (1/\mu(\mathcal{X}))\|h_1\|^2_{L^2(\mathcal{X},\lambda)}\|h_2\|^2_{L^2(\mathcal{X},\lambda)},$$

$$\mathbb{E}_\mu\left[\|f - \mathbb{E}_\mu[f|P_r]\|^2\right] = (1/\mu(\mathcal{X}))\|h_1\|^2_{L^2(\mathcal{X},\lambda)}\|h_2 - (1/\mu(\mathcal{X}))\int h_2 dx_2\|^2_{L^2(\mathcal{X},\lambda)}$$

$$= (1/\mu(\mathcal{X}))\|h_1\|^2_{L^2(\mathcal{X},\lambda)}\left(\|h_2\|^2_{L^2(\mathcal{X},\lambda)} - \frac{7}{16}\left(\int h_2 dx_2\right)^2\right),$$

where λ is the Lebesgue measure. □

6.4 Computational Complexity of Hierarchical top–down

In Table 1 we report the computational complexity of the hierarchical top–down clustering algorithm. We report the costs divided by level of refinement.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

References

1. Constantine, P.G.: Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies. SIAM Spotlights, vol. 2. SIAM, Pheliphedia (2015)
2. Tezzele, M., Romor, F., Rozza, G.: Reduction in parameter space, Chap. 16. In: Rozza, G., Stabile, G., Ballarin, F. (eds.) Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics. CS&E Series, SIAM Press, Pheliphedia (2022). https://doi.org/10.1137/1.9781611977257.ch16
3. Chinesta, F., Huerta, A., Rozza, G., Willcox, K.: Model reduction methods. In: Stein, E., de Borst, R., Hughes, T.J.R. (eds.) Encyclopedia of Computational Mechanics, 2nd edn., pp. 1–36. Wiley, Hoboken (2017)
4. Chinesta, F., Ladeveze, P., Cueto, E.: A short review on model order reduction based on proper generalized decomposition. Archiv. Comput. Methods Eng. **18**(4), 395 (2011). https://doi.org/10.1007/s11831-011-9064-7
5. Benner, P., Ohlberger, M., Patera, A., Rozza, G., Urban, K.: A short review on model order reduction based on proper generalized decomposition. MS&A Series, vol. 17. Springer, Berlin (2017)
6. Rozza, G., Hess, M., Stabile, G., Tezzele, M., Ballarin, F.: Basic ideas and tools for projection-based model reduction of parametric partial differential equations. In: Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W.H.A., Silveira, L.M. (eds.) Model Order Reduction, Chap. 1, pp. 1–47. De Gruyter, Berlin ( ). https://doi.org/10.1515/9783110671490-001
7. Brunton, S.L., Nathan Kutz, J., Manohar, K., Aravkin, A.Y., Morgansen, K., Klemisch, J., Goebel, N., Buttrick, J., Poskin, J., Blom-Schieber, A.W., et al.: Data-driven aerospace engineering: reframing the industry with machine learning. AIAA J. **59**(8), 2820–2847 (2021). https://doi.org/10.2514/1.J060131
8. Brunton, S.L., Kutz, J.N.: Data-driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge University Press, Cambridge (2019)
9. Rozza, G., Malik, M.H., Demo, N., Tezzele, M., Girfoglio, M., Stabile, G., Mola, A.: Advances in Reduced Order Methods for Parametric Industrial Problems in Computational Fluid Dynamics. In: Owen, R., de Borst, R., Reese, J., Chris, P. (eds.) Proceedings of the 6th European Conference on Computational Mechanics: Solids, Structures and Coupled Problems, ECCM 2018 and 7th European Conference on Computational Fluid Dynamics, ECFD 2018, Glasgow, UK, pp. 59–76 (2020)
10. Romor, F., Tezzele, M., Mrosek, M., Othmer, C., Rozza, G.: Multi-fidelity data fusion through parameter space reduction with applications to automotive engineering. Int. J. Numer. Meth. Eng. **124**(23), 5293–5311 (2023). https://doi.org/10.1002/nme.7349
11. Lukaczyk, T.W., Constantine, P., Palacios, F., Alonso, J.J.: Active subspaces for shape optimization. In: 10th AIAA Multidisciplinary Design Optimization Conference, p. 1171 (2014)
12. Tezzele, M., Salmoiraghi, F., Mola, A., Rozza, G.: Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape design problems. Adv. Model. Simul. Eng. Sci. **5**(1), 25 (2018). https://doi.org/10.1186/s40323-018-0118-3
13. Boncoraglio, G., Farhat, C., Bou-Mosleh, C.: Model reduction framework with a new take on active subspaces for optimization problems with linearized fluid-structure interaction constraints. Int. J. Numer. Meth. Eng. (2020). https://doi.org/10.1002/nme.6376

14. Demo, N., Tezzele, M., Rozza, G.: A supervised learning approach involving active subspaces for an efficient genetic algorithm in high-dimensional optimization problems. SIAM J. Sci. Comput. **43**(3), 831–853 (2021). https://doi.org/10.1137/20M1345219

15. Demo, N., Tezzele, M., Mola, A., Rozza, G.: Hull shape design optimization with parameter space and model reductions, and self-learning mesh morphing. J. Mar. Sci. Eng. **9**(2), 185 (2021). https://doi.org/10.3390/jmse9020185

16. Zahm, O., Cui, T., Law, K., Spantini, A., Marzouk, Y.: Certified dimension reduction in nonlinear Bayesian inverse problems. Math. Comput. **91**(336), 1789–1835 (2022). https://doi.org/10.1090/mcom/3737

17. Cortesi, A.F., Constantine, P.G., Magin, T.E., Congedo, P.M.: Forward and backward uncertainty quantification with active subspaces: application to hypersonic flows around a cylinder. J. Comput. Phys. **407**, 109079 (2020). https://doi.org/10.1016/j.jcp.2019.109079

18. Tezzele, M., Ballarin, F., Rozza, G.: Combined parameter and model reduction of cardiovascular problems by means of active subspaces and POD-Galerkin methods. In: Boffi, D., Pavarino, L.F., Rozza, G., Scacchi, S., Vergara, C. (eds.) Mathematical and Numerical Modeling of the Cardiovascular System and Applications. SEMA-SIMAI Series, vol. 16, pp. 185–207. Springer, Berlin (2018). https://doi.org/10.1007/978-3-319-96649-6_8

19. Demo, N., Tezzele, M., Rozza, G.: A non-intrusive approach for reconstruction of POD modal coefficients through active subspaces. C. R. Mécanique de l'Académie des Sci. **347**(11), 873–881 (2019). https://doi.org/10.1016/j.crme.2019.11.012

20. Tezzele, M., Fabris, L., Sidari, M., Sicchiero, M., Rozza, G.: A multi-fidelity approach coupling parameter space reduction and non-intrusive POD with application to structural optimization of passenger ship hulls. Int. J. Numer. Meth. Eng. **124**(5), 1193–1210 (2023). https://doi.org/10.1002/nme.7159

21. Romor, F., Tezzele, M., Lario, A., Rozza, G.: Kernel-based active subspaces with application to computational fluid dynamics parametric problems using discontinuous Galerkin method. Int. J. Numer. Meth. Eng. **123**(23), 6000–6027 (2022). https://doi.org/10.1002/nme.7099

22. Zhang, G., Zhang, J., Hinkle, J.: Learning nonlinear level sets for dimensionality reduction in function approximation. In: Advances in Neural Information Processing Systems, pp. 13199–13208 (2019)

23. Bridges, R.A., Gruber, A.D., Felder, C.R., Verma, M., Hoff, C.: Active Manifolds: A non-linear analogue to Active Subspaces. In: Proceddings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA, pp. 764–772 (2019)

24. Chen, P., Ghattas, O.: Hessian-based sampling for high-dimensional model reduction. Int. J. Uncertain. Quant. (2019). https://doi.org/10.1615/Int.J.UncertaintyQuantification.2019028753

25. Løvgren, A.E., Maday, Y., Rønquist, E.M.: A reduced basis element method for the steady Stokes problem. ESAIM Math. Model. Numer. Anal. **40**(3), 529–552 (2006). https://doi.org/10.1051/m2an:2006021

26. Amsallem, D., Farhat, C.: Interpolation method for adapting reduced-order models and application to aeroelasticity. AIAA J. **46**(7), 1803–1813 (2008). https://doi.org/10.2514/1.35374

27. Daniel, T., Casenave, F., Akkari, N., Ryckelynck, D.: Model order reduction assisted by deep neural networks (ROM-net). Adv. Model. Simul. Eng. Sci. **7**(1), 1–27 (2020). https://doi.org/10.1186/s40323-020-00153-6

28. Wu, Q., Liang, F., Mukherjee, S.: Localized sliced inverse regression. J. Comput. Graph. Stat. **19**(4), 843–860 (2010). https://doi.org/10.1198/jcgs.2010.08080

29. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. IEEE Trans. Pattern Anal. Mach. Intell. **18**(6), 607–616 (1996). https://doi.org/10.1109/34.506411

30. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. J. Mach. Learn. Res. **8**(5), 1027–1061 (2007)

31. Li, K.-C.: Sliced inverse regression for dimension reduction. J. Am. Stat. Assoc. **86**(414), 316–327 (1991). https://doi.org/10.2307/2290563

32. Xiong, J., Cai, X., Li, J.: Clustered active-subspace based local gaussian process emulator for high-dimensional and complex computer models. J. Comput. Phys. **450**, 110840 (2022). https://doi.org/10.1016/j.jcp.2021.110840

33. Sullivan, T.J.: Introduction to Uncertainty Quantification, vol. 63. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23395-6

34. Zahm, O., Constantine, P.G., Prieur, C., Marzouk, Y.M.: Gradient-based dimension reduction of multivariate vector-valued functions. SIAM J. Sci. Comput. **42**(1), 534–558 (2020). https://doi.org/10.1137/18M1221837

35. Pinkus, A.: Ridge Functions, vol. 205. Cambridge University Press, Cambridge (2015)

36. Parente, M.T., Wallin, J., Wohlmuth, B., et al.: Generalized bounds for active subspaces. Electron. J. Stat. **14**(1), 917–943 (2020). https://doi.org/10.1214/20-EJS1684

37. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. The Morgan Kaufm. Ser. Data Manage. Syst. **5**(4), 83–124 (2012). https://doi.org/10.1016/C2009-0-61819-5

38. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Series in Probability and Statistics, vol. 344. Wiley, Hoboken (2005)
39. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge (2012)
40. Park, H.-S., Jun, C.-H.: A simple and fast algorithm for K-medoids clustering. Expert Syst. Appl. **36**(2), 3336–3341 (2009). https://doi.org/10.1016/j.eswa.2008.01.039
41. Schubert, E., Rousseeuw, P.J.: Faster k-medoids clustering: improving the PAM, CLARA, and CLARANS algorithms. In: International Conference on Similarity Search and Applications, pp. 171–187 (2019). https://doi.org/10.1007/978-3-030-32047-8_16. Springer
42. Maranzana, F.E.: On the location of supply points to minimize transport costs. J. Oper. Res. Soc. **15**(3), 261–270 (1964). https://doi.org/10.1057/jors.1964.47
43. Williams, C.K., Rasmussen, C.E.: Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning Series, MIT Press, Cambridge (2006)
44. Romor, F., Tezzele, M., Rozza, G.: ATHENA: Advanced techniques for high dimensional parameter spaces to enhance numerical analysis. Softw. Impacts **10**, 100133 (2021). https://doi.org/10.1016/j.simpa.2021.100133
45. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122 (2013)
46. GPy: GPy: A Gaussian process framework in Python. http://github.com/SheffieldML/GPy (since 2012)
47. Diaz, P., Constantine, P., Kalmbach, K., Jones, E., Pankavich, S.: A modified SEIR model for the spread of Ebola in Western Africa and metrics for resource allocation. Appl. Math. Comput. **324**, 141–155 (2018). https://doi.org/10.1016/j.amc.2017.11.039
48. Tezzele, M., Demo, N., Stabile, G., Mola, A., Rozza, G.: Enhancing CFD predictions in shape design problems by model and parameter space reduction. Adv. Model. Simul. Eng. Sci. (2020). https://doi.org/10.1186/s40323-020-00177-y
49. Abbott, I.H., Von Doenhoff, A.E.: Theory of Wing Sections: Including a Summary of Airfoil Data. Courier Corporation, North Chelmsford (2012)
50. Jacobs, E.N., Ward, K.E., Pinkerton, R.M.: The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. Technical Report 430, N.A.C.A. (1933)
51. Hicks, R.M., Henne, P.A.: Wing design by numerical optimization. J. Aircr. **15**(7), 407–412 (1978). https://doi.org/10.2514/3.58379
52. Tropp, J.A.: User-friendly tail bounds for sums of random matrices. Found. Comput. Math. **12**(4), 389–434 (2012). https://doi.org/10.1007/s10208-011-9099-z
53. Golub, G.H., Van Loan, C.F.: Matrix Computations, vol. 3. Johns Hopkins University Press, Baltimore (2013)
54. Basu, K., Owen, A.B.: Transformations and Hardy–Krause Variation. SIAM J. Numer. Anal. **54**(3), 1946–1966 (2016). https://doi.org/10.1137/15M1052184