ETICS research school
3-7 oct. 2022

High-Dimensional Approximation

Part 3: Approximation from samples

Anthony Nouy

Centrale Nantes, Nantes Université, Laboratoire de Mathématiques Jean Leray

Approximation from limited information

Consider the approximation of functions $f : \mathcal{X} \to \mathbb{R}$ from a set $K \subset X$ using n information

$$\ell_1(f), \ldots, \ell_n(f)$$

that can be deterministic or random.

When $\ell_i : X \to \mathbb{R}$ are linear (or affine) maps, we talk about linear (or affine) information.

Type of information

A particular type of linear information is point evaluations (aka standard information)

$$\ell_i(f) = f(x_i)$$

Another type of linear information is

$$\ell_i(f) = \int_{\mathcal{X}} \psi_i(x) f(x) d\mu(x)$$

Type of information

A particular type of linear information is point evaluations (aka standard information)

$$\ell_i(f) = f(x_i)$$

Another type of linear information is

$$\ell_i(f) = \int_{\mathcal{X}} \psi_i(x) f(x) d\mu(x)$$

If f is known to satisfy an equation

$$B(f) = b$$

with given right-hand side $b \in Z' \subset \mathbb{R}^{\mathcal{X}}$ and operator $B : X \to Z'$, we can have access to the information

$$\ell_i(f) = B(f)(x_i), \quad \text{or} \quad \ell_i(f) = \langle \psi_i, B(f) \rangle$$

for some function $\psi_i \in Z$. For linear (resp. nonlinear) operator B, this corresponds to linear (resp. nonlinear) information. This is the framework of Galerkin or variational methods for PDEs, Physics-informed machine learning (Deep-Galerkin, Deep-Ritz, PINN, ...).

Type of information

We distinguish two different settings

- information is given (passive learning)
- information can be freely generated (active learning), a typical setting in computer/physical experiments, numerical analysis of PDEs, or scientific machine learning.

Algorithm

Given information $\ell(f) = (\ell_1(f), \ldots, \ell_n(f))$, an algorithm returns an approximation

$$A(\ell(f))$$

in a subset of X, where the map A is related to the choice of a restricted model class (or approximation tool).

A linear algorithm, with A also a linear map, corresponds to linear approximation:

$$A(\ell(f)) = \sum_{i=1}^{n} a_i(\ell(f))\varphi_i$$

where the a_i are linear maps and $span\{\varphi_1, \ldots, \varphi_n\}$ is the range of A.

Restricted model classes

The approximation problem from limited information is an ill-posed problem unless some additional information on the function class K is taken into account.

It could be a low-dimensional manifold V_m (model class) that is known to approximate well the set K, or a sequence of models with increasing complexity $(V_m)_{m\geq 1}$ (approximation tool) that is known to approximate the manifold with a good rate of convergence.

Restricted model classes

The approximation problem from limited information is an ill-posed problem unless some additional information on the function class K is taken into account.

It could be a low-dimensional manifold V_m (model class) that is known to approximate well the set K, or a sequence of models with increasing complexity $(V_m)_{m \geq 1}$ (approximation tool) that is known to approximate the manifold with a good rate of convergence.

- For K a ball of Sobolev or Besov spaces: splines (with fixed or adaptive mesh) or wavelets (with or without sparsity)

Restricted model classes

The approximation problem from limited information is an ill-posed problem unless some additional information on the function class K is taken into account.

It could be a low-dimensional manifold V_m (model class) that is known to approximate well the set K, or a sequence of models with increasing complexity $(V_m)_{m\geq 1}$ (approximation tool) that is known to approximate the manifold with a good rate of convergence.

- For K a ball of Sobolev or Besov spaces: splines (with fixed or adaptive mesh) or wavelets (with or without sparsity)
- For K a set of analytic functions: polynomial spaces

Restricted model classes

The approximation problem from limited information is an ill-posed problem unless some additional information on the function class K is taken into account.

It could be a low-dimensional manifold V_m (model class) that is known to approximate well the set K, or a sequence of models with increasing complexity $(V_m)_{m\geq 1}$ (approximation tool) that is known to approximate the manifold with a good rate of convergence.

- For K a ball of Sobolev or Besov spaces: splines (with fixed or adaptive mesh) or wavelets (with or without sparsity)
- For K a set of analytic functions: polynomial spaces
- For K a set of analytic functions with singularities: rational polynomials, h-p splines

Restricted model classes

The approximation problem from limited information is an ill-posed problem unless some additional information on the function class K is taken into account.

It could be a low-dimensional manifold V_m (model class) that is known to approximate well the set K, or a sequence of models with increasing complexity $(V_m)_{m \geq 1}$ (approximation tool) that is known to approximate the manifold with a good rate of convergence.

- For K a ball of Sobolev or Besov spaces: splines (with fixed or adaptive mesh) or wavelets (with or without sparsity)
- For K a set of analytic functions: polynomial spaces
- For K a set of analytic functions with singularities: rational polynomials, h-p splines
- For a larger class of sets K: neural networks or tensor networks

Restricted model classes

The approximation problem from limited information is an ill-posed problem unless some additional information on the function class K is taken into account.

It could be a low-dimensional manifold V_m (model class) that is known to approximate well the set K, or a sequence of models with increasing complexity $(V_m)_{m \geq 1}$ (approximation tool) that is known to approximate the manifold with a good rate of convergence.

- For K a ball of Sobolev or Besov spaces: splines (with fixed or adaptive mesh) or wavelets (with or without sparsity)
- For K a set of analytic functions: polynomial spaces
- For K a set of analytic functions with singularities: rational polynomials, h-p splines
- For a larger class of sets K: neural networks or tensor networks
- For more general manifolds K, V_m can be obtained by manifold approximation (or dimension reduction) methods

Approximation in a given model class

For a given model class V_m and given information $z = \ell(f)$, an approximation $f_m = A(z) \in V_m$ may be defined by

$$\ell(f_m) = z. \tag{1}$$

If for any z there exists a unique element f_m in V_m satisfying (1), we say that ℓ is unisolvent for V_m. When information are point evaluations, this corresponds to interpolation. When information are linear functionals of an equation's residual, this corresponds to (Petrov-)Galerkin projection.

Approximation in a given model class

For a given model class V_m and given information $z = \ell(f)$, an approximation $f_m = A(z) \in V_m$ may be defined by

$$\ell(f_m) = z. \tag{1}$$

If for any z there exists a unique element f_m in V_m satisfying (1), we say that ℓ is unisolvent for V_m. When information are point evaluations, this corresponds to interpolation. When information are linear functionals of an equation's residual, this corresponds to (Petrov-)Galerkin projection.

More generally, $f_m = A(z)$ can be defined as a solution of

$$\min_{f_m \in V_m} d(\ell(f_m), z),$$

and in particular

$$\min_{f_m \in V_m} \sum_{i=1}^{n} w_i (\ell_i(f_m) - z_i)^2$$

When information are point evaluations, this corresponds to (weighted) least-squares approximation. When information are linear functionals of an equation's residual, this corresponds to (Petrov-)Galerkin projection.

Adaptivity, model selection

When the information is given (passive learning), the complexity of the model class V_m is limited. Adaptive strategies play with a collection of model classes $(V_m)_{m\geq 1}$ and require model selection techniques to take the best from the available information.

Adaptivity, model selection

When the information is given (passive learning), the complexity of the model class V_m is limited. Adaptive strategies play with a collection of model classes $(V_m)_{m\geq 1}$ and require model selection techniques to take the best from the available information.

When the information can be generated (active learning), a fundamental question is how to generate a good information for a given model class V_m. Adaptive strategies play with a collection of model classes $(V_m)_{m\geq 1}$ and generate information adaptively. A question is then to recycle information in order to obtain a near-optimal performance in terms of complexity.

Outline

1. Manifold approximation
2. Linear approximation from point evaluations
3. Tensor networks approximation with point evaluations

Outline

1. Manifold approximation
2. Linear approximation from point evaluations
3. Tensor networks approximation with point evaluations

Manifold approximation

Assume we want to approximate (or recover) functions from a general manifold K in a vector space X. If K can be sampled, a suitable low-dimensional model class V_m (or sequence of model classes) can be obtained by manifold approximation (or dimension reduction) methods using samples from K.

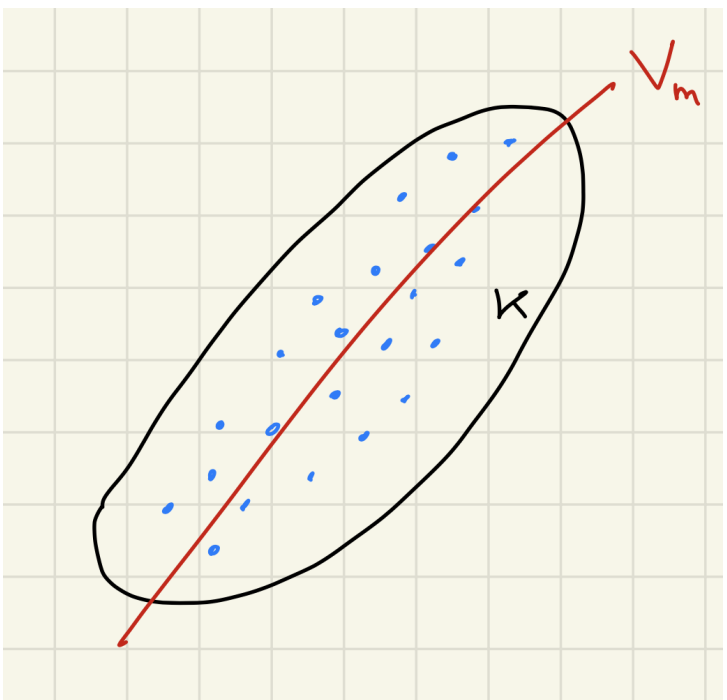
Manifold approximation

Assume we want to approximate (or recover) functions from a general manifold K in a vector space X. If K can be sampled, a suitable low-dimensional model class V_m (or sequence of model classes) can be obtained by manifold approximation (or dimension reduction) methods using samples from K.

Typical model classes V_m include

- Low-dimensional linear/affine spaces

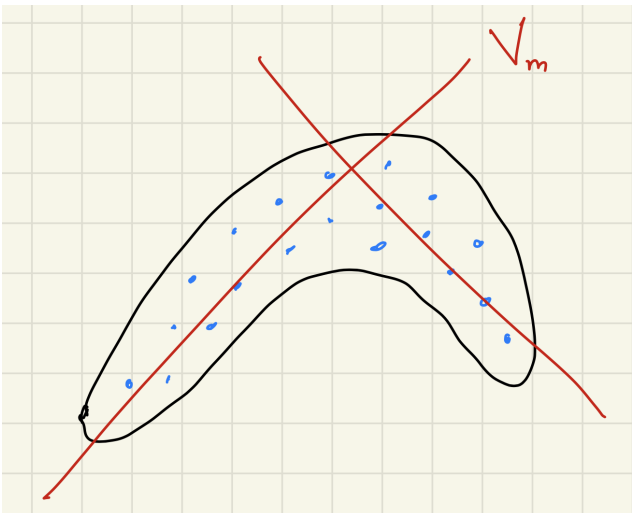
$$V_m = \{g(a) : a \in \mathbb{R}^m\}, \quad \text{with } g : \mathbb{R}^m \to X \text{ linear/affine}$$



Manifold approximation

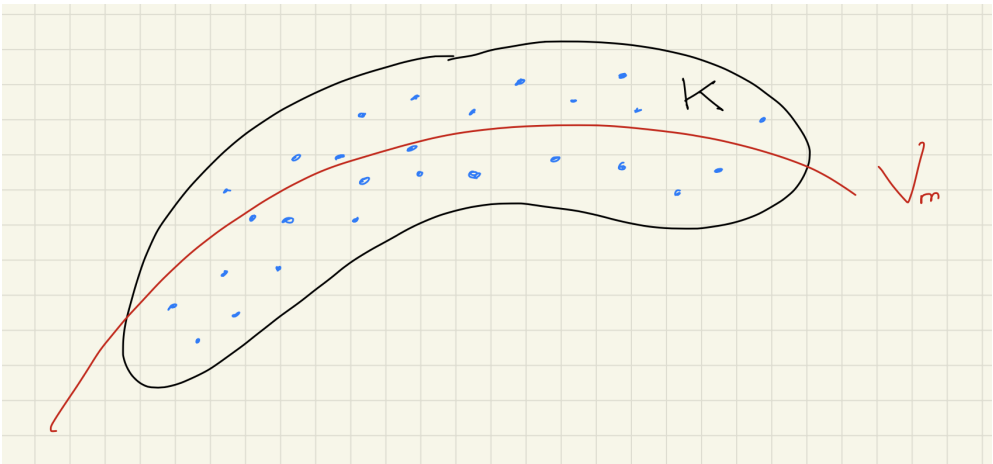
- Union of low-dimensional linear spaces

$$V_m = \bigcup_{k=1}^{m} W_k$$



Manifold approximation

- Manifold $V_m = \{g(a) : a \in \mathbb{R}^m\}$ with continuous parametrization map $g : \mathbb{R}^m \to X$.



Manifold approximation

A typical setting is when K is the set of trajectories of a random process or more generally the range of some function-valued random variable. A possible dimension reduction method is principal component analysis (for linear approximation).

Another setting is the solution of forward or inverse problems of parameter-dependent equations where $K = \{u(y) : y \in Y\}$ is the manifold of solutions. Manifold approximation is called model order reduction (reduced basis, POD, ...).

Principal component analysis (for linear approximation)

Let Y be equipped with a probability measure μ and X a Hilbert space, and $K = \{u(y) : y \in Y\}$ with u a map in the Bochner space $L^2(Y; X)$.

The optimal performance of a linear approximation of K is measured in mean-squared error by

$$d_m^{2,\mu}(K)_X = \inf_{\dim(V_m)=m} \int_Y E(u(y); V_m)_X^2 \, d\mu(y) = \inf_{\dim(V_m)=m} \mathbb{E}_{y\sim\mu}(\|u(y) - P_{V_m} u(y)\|_X^2)$$

Principal component analysis (for linear approximation)

Let Y be equipped with a probability measure μ and X a Hilbert space, and $K = \{u(y) : y \in Y\}$ with u a map in the Bochner space $L^2(Y; X)$.

The optimal performance of a linear approximation of K is measured in mean-squared error by

$$d_m^{2,\mu}(K)_X = \inf_{\dim(V_m)=m} \int_Y E(u(y); V_m)_X^2 \, d\mu(y) = \inf_{\dim(V_m)=m} \mathbb{E}_{y\sim\mu}(\|u(y) - P_{V_m} u(y)\|_X^2)$$

An optimal subspace V_m is given by principal component analysis (PCA), where V_m is the dominant eigenspace of the self-adjoint compact operator $T : v \mapsto \mathbb{E}_{y\sim\mu}((u(y), v)_X u(y))$ and the error is

$$\mathbb{E}_{y\sim\mu}(\|u(y) - P_{V_m} u(y)\|_X^2) = \sum_{i>m} \lambda_i$$

where $(\lambda_i)_{i\geq 1}$ is the decreasing sequence of eigenvalues of T. This is related to singular value decomposition (or Karhunen-Loeve decomposition) of $u \in L^2(Y) \otimes X$,

$$u(y) = \sum_{i\geq 1} \sqrt{\lambda_i} \varphi_i a_i(y), \quad P_{V_m} u(y) = \sum_{i=1}^{m} \sqrt{\lambda_i} \varphi_i a_i(y)$$

Principal component analysis (for linear approximation)

Let Y be equipped with a probability measure μ and X a Hilbert space, and $K = \{u(y) : y \in Y\}$ with u a map in the Bochner space $L^2(Y; X)$.

The optimal performance of a linear approximation of K is measured in mean-squared error by

$$d_m^{2,\mu}(K)_X = \inf_{\dim(V_m)=m} \int_Y E(u(y); V_m)_X^2 \, d\mu(y) = \inf_{\dim(V_m)=m} \mathbb{E}_{y\sim\mu}(\|u(y) - P_{V_m} u(y)\|_X^2)$$

An optimal subspace V_m is given by principal component analysis (PCA), where V_m is the dominant eigenspace of the self-adjoint compact operator $T : v \mapsto \mathbb{E}_{y\sim\mu}((u(y), v)_X u(y))$ and the error is

$$\mathbb{E}_{y\sim\mu}(\|u(y) - P_{V_m} u(y)\|_X^2) = \sum_{i>m} \lambda_i$$

where $(\lambda_i)_{i\geq 1}$ is the decreasing sequence of eigenvalues of T. This is related to singular value decomposition (or Karhunen-Loeve decomposition) of $u \in L^2(Y) \otimes X$,

$$u(y) = \sum_{i\geq 1} \sqrt{\lambda_i} \varphi_i a_i(y), \quad P_{V_m} u(y) = \sum_{i=1}^{m} \sqrt{\lambda_i} \varphi_i a_i(y)$$

PCA even provides a hierarchical sequence of model classes $(V_m)_{m\geq 1}$.

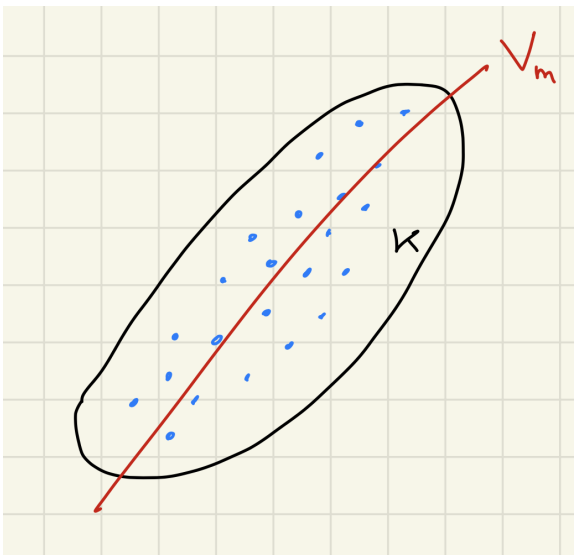
Principal component analysis (for linear approximation)

An estimation of V_m is given by empirical PCA which consists in solving

$$\min_{\dim(V_m)=m} \frac{1}{n} \sum_{i=1}^{n} \|u(y_i) - P_{V_m} u(y_i)\|_X^2$$

where the y_i are samples in Y and the $u(y_i)$ are the corresponding samples in K.
The solution is the dominant eigenspace of the operator

$$T_n : v \mapsto \frac{1}{n} \sum_{i=1}^{n} u(y_i)(u(y_i), v)_X.$$



For an analysis of empirical PCA, see e.g. [3, 4].

Principal component analysis (for linear approximation)

Assuming X is finite dimensional with orthonormal basis $(e_i)_{1\leq i\leq N}$, $u(y) = \sum_{i=1}^{N} a_i(y)e_i$, and a basis of V_m is given by the dominant eigenvectors of the matrix

$$\frac{1}{n}\sum_{i=1}^{n} a(y_i)a(y_i)^T.$$

This is equivalent to obtain the dominant left singular vectors of the matrix

$$A = (a(y_1), \dots a(y_n)) \in \mathbb{R}^{N\times n}$$

Optimal sampling strategy have been proposed for singular value decomposition of matrices. This requires an estimation of dominant right singular vectors.

Greedy algorithms (for linear approximation)

Given a set K from a Banach space X, the optimal performance of linear approximation in worst case error is measured through the Kolmogorov width

$$d_n(K)_X = \inf_{\dim(V_m)=m} \sup_{u\in K} E(u, V_m) \quad \text{with} \quad E(u, V_m)_X := \inf_{v\in V_m} \|u - v\|_X$$

Greedy algorithms (for linear approximation)

Given a set K from a Banach space X, the optimal performance of linear approximation in worst case error is measured through the Kolmogorov width

$$d_n(K)_X = \inf_{\dim(V_m)=m} \sup_{u\in K} E(u, V_m) \quad \text{with} \quad E(u, V_m)_X := \inf_{v\in V_m} \|u - v\|_X$$

Greedy algorithms can be used to the construction of a hierarchical sequence of spaces $(V_m)_{m\geq 1}$ using samples (snapshots) from K. Spaces are defined by $V_m = span\{u_1, \ldots, u_m\}$ where $(u_i)_{i\geq 1}$ is a sequence from K selected greedily.

Greedy algorithms (for linear approximation)

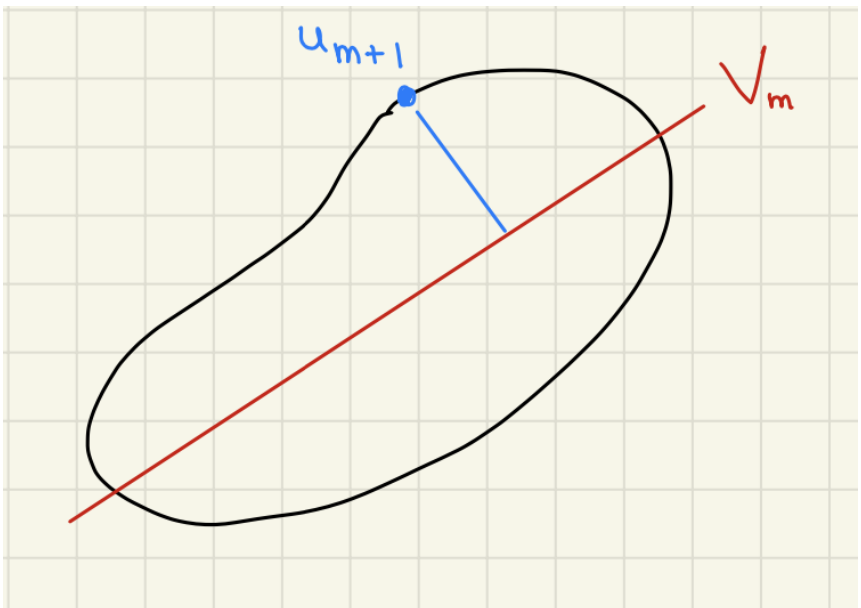
Given a set K from a Banach space X, the optimal performance of linear approximation in worst case error is measured through the Kolmogorov width

$$d_n(K)_X = \inf_{\dim(V_m)=m} \sup_{u\in K} E(u, V_m) \quad \text{with} \quad E(u, V_m)_X := \inf_{v\in V_m} \|u - v\|_X$$

Greedy algorithms can be used to the construction of a hierarchical sequence of spaces $(V_m)_{m\geq 1}$ using samples (snapshots) from K. Spaces are defined by $V_m = span\{u_1, \ldots, u_m\}$ where $(u_i)_{i\geq 1}$ is a sequence from K selected greedily.

Given V_m, u_{m+1} is the element which provides the highest error of approximation by V_m

$$E(u_{m+1}, V_m)_X = \max_{u\in K} E(u, V_m)_X$$



Greedy algorithms (for linear approximation)

When $K = \{u(y) : y \in Y\}$, $u_{m+1} = u(y_{m+1})$ where the parameter value y_{m+1} is such that

$$y_{m+1} \in \arg\max_{y \in Y} E(u(y), V_m)_X$$

In practice, for a computationally feasible algorithm, $E(u(y), V_m)_X$ is replaced by some error estimate $\Delta(u(y), V_m)$, and the maximum is taken over a finite training set in Y (possibly random [Cohen et al 2020]).

Greedy algorithms (for linear approximation)

When $K = \{u(y) : y \in Y\}$, $u_{m+1} = u(y_{m+1})$ where the parameter value y_{m+1} is such that

$$y_{m+1} \in \arg\max_{y \in Y} E(u(y), V_m)_X$$

In practice, for a computationally feasible algorithm, $E(u(y), V_m)_X$ is replaced by some error estimate $\Delta(u(y), V_m)$, and the maximum is taken over a finite training set in Y (possibly random [Cohen et al 2020]).

A typical setting is when $K = \{u(y) : y \in Y\} \subset X$ is the solution of some parameter dependent equation

$$R(u(y); y) = 0$$

Here $\Delta(u(y), V_m)$ is typically defined as some residual norm

$$\Delta(u(y), V_m) = \|R(u_m(y); y)\|$$

with $u_m(y)$ a Galerkin projection of $u(y)$ onto V_m.

Greedy algorithms (for linear approximation)

When $K = \{u(y) : y \in Y\}$, $u_{m+1} = u(y_{m+1})$ where the parameter value y_{m+1} is such that

$$y_{m+1} \in \arg\max_{y \in Y} E(u(y), V_m)_X$$

In practice, for a computationally feasible algorithm, $E(u(y), V_m)_X$ is replaced by some error estimate $\Delta(u(y), V_m)$, and the maximum is taken over a finite training set in Y (possibly random [Cohen et al 2020]).

A typical setting is when $K = \{u(y) : y \in Y\} \subset X$ is the solution of some parameter dependent equation

$$R(u(y); y) = 0$$

Here $\Delta(u(y), V_m)$ is typically defined as some residual norm

$$\Delta(u(y), V_m) = \|R(u_m(y); y)\|$$

with $u_m(y)$ a Galerkin projection of $u(y)$ onto V_m.

Randomized linear algebra can be used for an efficient and stable estimation of residual norms [Balabanov and Nouy 2021a], and for the construction of preconditioners [Balabanov and Nouy 2021b].

Greedy algorithms (for linear approximation)

This yield a suboptimal selection of u_{m+1} satisfying

$$E(u_{m+1}, V_m)_X \geq \gamma \max_{u \in K} E(u, V_m)_X, \quad \gamma \leq 1.$$

This algorithm therefore generates a suboptimal sequence of spaces yielding a worst case error

$$\sigma_m(K)_X := \sup_{u \in K} E(u, V_m)_X \geq d_m(K)_X$$

Greedy algorithms (for linear approximation)

This yield a suboptimal selection of u_{m+1} satisfying

$$E(u_{m+1}, V_m)_X \geq \gamma \max_{u \in K} E(u, V_m)_X, \quad \gamma \leq 1.$$

This algorithm therefore generates a suboptimal sequence of spaces yielding a worst case error

$$\sigma_m(K)_X := \sup_{u \in K} E(u, V_m)_X \geq d_m(K)_X$$

Assuming $\gamma \geq 1$ is independent of m, the algorithm is a weak greedy algorithm for which results have been obtained in [DeVore et al 2013].

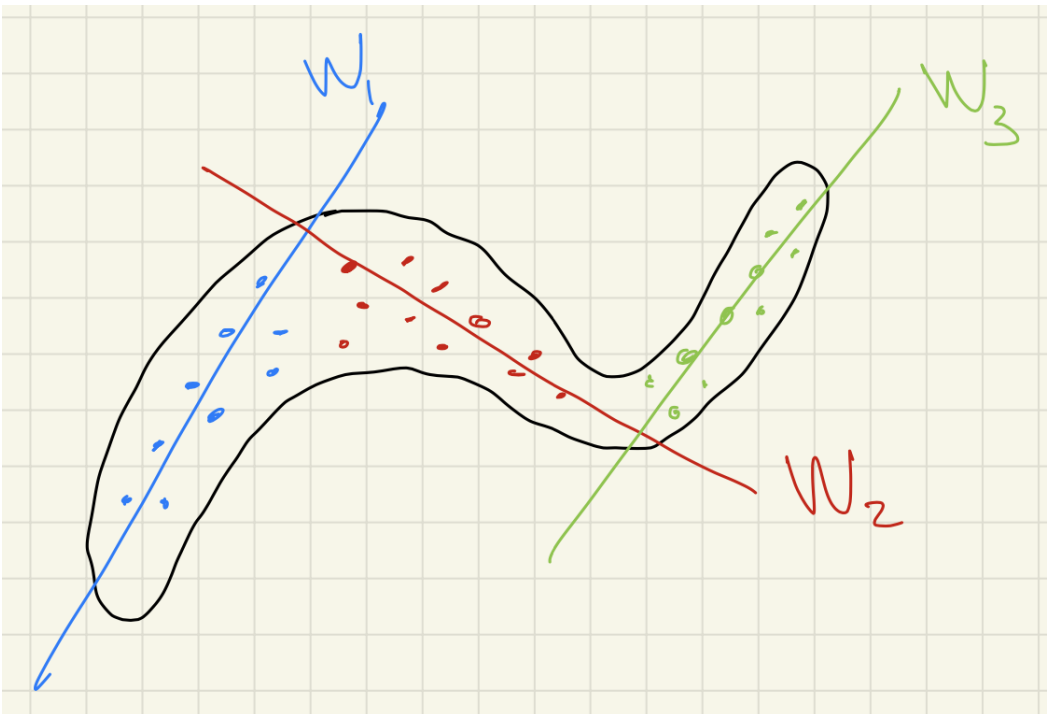
For X a Hilbert space, it holds

- $\sigma_{2m}(K)_X \leq \sqrt{2}\gamma^{-1}\sqrt{d_m(K)_X}$
- If $d_m(K)_X \leq C_0 m^{-\alpha}$ then $\sigma_m(K)_X \leq C_1 m^{-\alpha}$
- If $d_m(K)_X \leq C_0 e^{-c_0 m^{\alpha}}$ then $\sigma_m(K)_X \leq C_1 e^{-c_1 m^{\alpha}}$

For X a Banach space, similar but slightly worse results hold.

Multi-space approximation

h or h-p reduced basis methods [Eftang et al 2010] are multi-space approximation methods that consist is partitioning the manifold K (or corresponding parameter set Y) into subsets K_k, and approximating each subset by a linear space W_k of fixed dimension (h method) or variable dimension (h-p method).

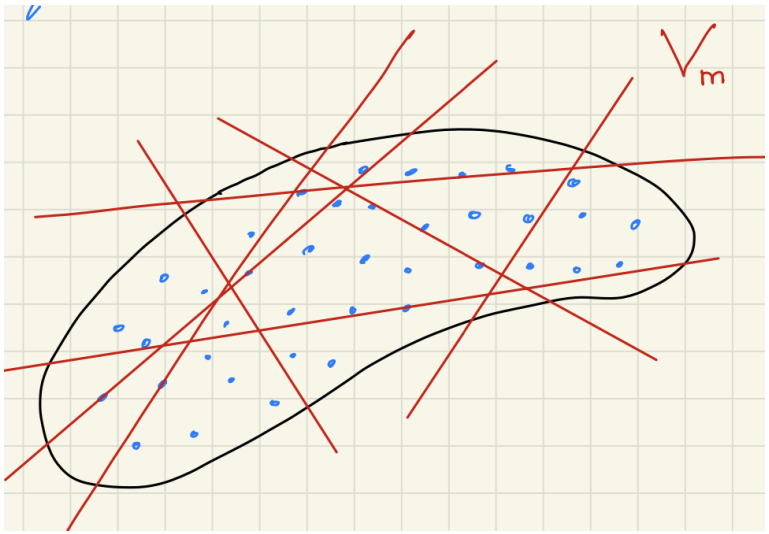


These methods requires a partitioning (or clustering) strategy.

Dictionary-based multi-space approximation

Multiple spaces can be extracted from a dictionary $\mathcal{D} = \{u_1, \ldots, u_N\}$ of samples from K. By considering subspaces with dimension less than m, this yields the model class

$$V_m := V_m(\mathcal{D}) = \bigcup_{\alpha \in \{1,\ldots,N\}^m} W_\alpha(\mathcal{D}), \quad W_\alpha(\mathcal{D}) = span\{u_{\alpha_1}, \ldots, u_{\alpha_m}\}$$



This is equivalent to m-term approximation

$$V_m = \{g(a) := \sum_{i=1}^{N} a_i u_i : a \in \mathbb{R}^N, \|a\|_0 \leq m\}.$$

The dictionary (samples) can be taken arbitrarily or generated with a greedy procedure proposed in [Balabanov and Nouy 2021a], using randomized linear algebra for handling large dictionaries.

Nonlinear manifold approximation

Several approaches exist for the approximation of a set K by a parametrized nonlinear manifold of the form

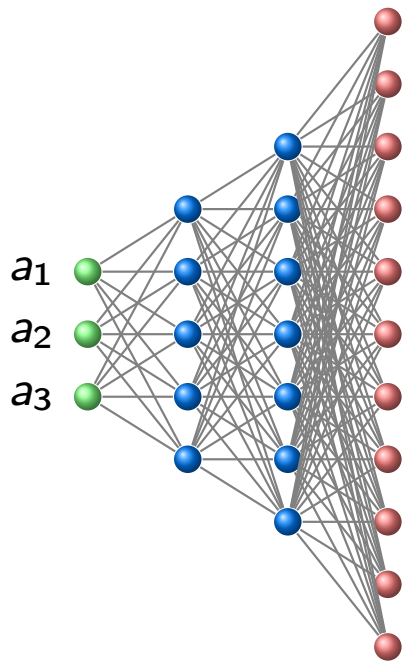
$$V_m = \{g(a) : a \in \mathbb{R}^m\}, \quad g : \mathbb{R}^m \to X.$$

Nonlinear manifold approximation

Several approaches exist for the approximation of a set K by a parametrized nonlinear manifold of the form

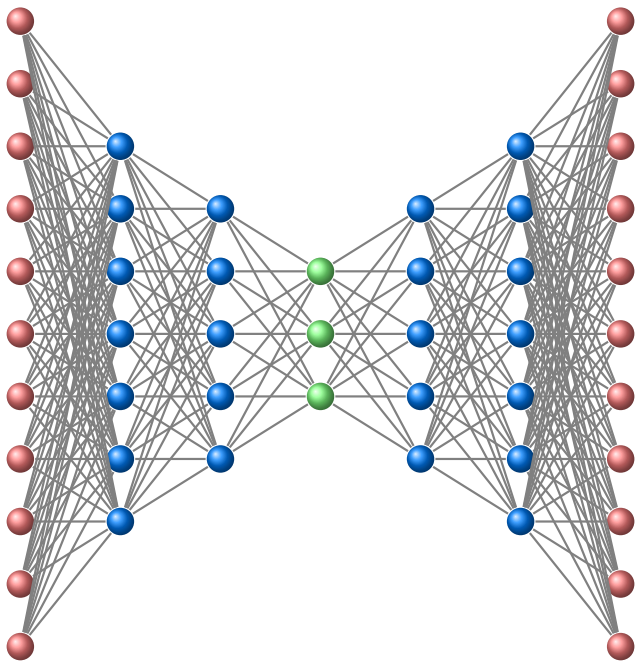
$$V_m = \{g(a) : a \in \mathbb{R}^m\}, \quad g : \mathbb{R}^m \to X.$$

Neural networks are popular tools for this task.
For $X = \mathbb{R}^N$, a neural network representation can be used for $g : \mathbb{R}^m \to \mathbb{R}^N$.



Nonlinear manifold approximation

Learning a map g from samples from K can be done (offline) by learning a compositional function (or autoencoder) $g \circ h$, where both functions $h : \mathbb{R}^N \to \mathbb{R}^m$ (the encoder) and $g : \mathbb{R}^m \to \mathbb{R}^N$ (the decoder) can be represented by neural networks.



Nonlinear manifold approximation

Given samples $\{u_1, \ldots, u_n\} \subset K$, h and g can be obtained by minimizing

$$\sum_{i=1}^{n} \|u_i - g \circ h(u_i)\|_X^2 \tag{2}$$

Nonlinear manifold approximation

Given samples $\{u_1, \ldots, u_n\} \subset K$, h and g can be obtained by minimizing

$$\sum_{i=1}^{n} \|u_i - g \circ h(u_i)\|_X^2 \tag{2}$$

This methodology is not restricted to the use of neural networks for h and g.

For h, one can use a linear map (a matrix of size $N \times m$), so that $g \circ h$ corresponds to a ridge approximation.

Note that if h and g are restricted to be linear maps (or matrices of size $N \times m$ and $m \times N$ respectively), it boils down to linear approximation learned by PCA.

Nonlinear manifold approximation

Given samples $\{u_1, \ldots, u_n\} \subset K$, h and g can be obtained by minimizing

$$\sum_{i=1}^{n} \|u_i - g \circ h(u_i)\|_X^2 \tag{2}$$

This methodology is not restricted to the use of neural networks for h and g.

For h, one can use a linear map (a matrix of size $N \times m$), so that $g \circ h$ corresponds to a ridge approximation.

Note that if h and g are restricted to be linear maps (or matrices of size $N \times m$ and $m \times N$ respectively), it boils down to linear approximation learned by PCA.

A two-step strategy can be used, by first learning a composition of linear maps $\tilde{g} \circ h$ by PCA, or another algorithm for linear approximation, and then learning $g \circ h$ with a fixed h by solving (2).

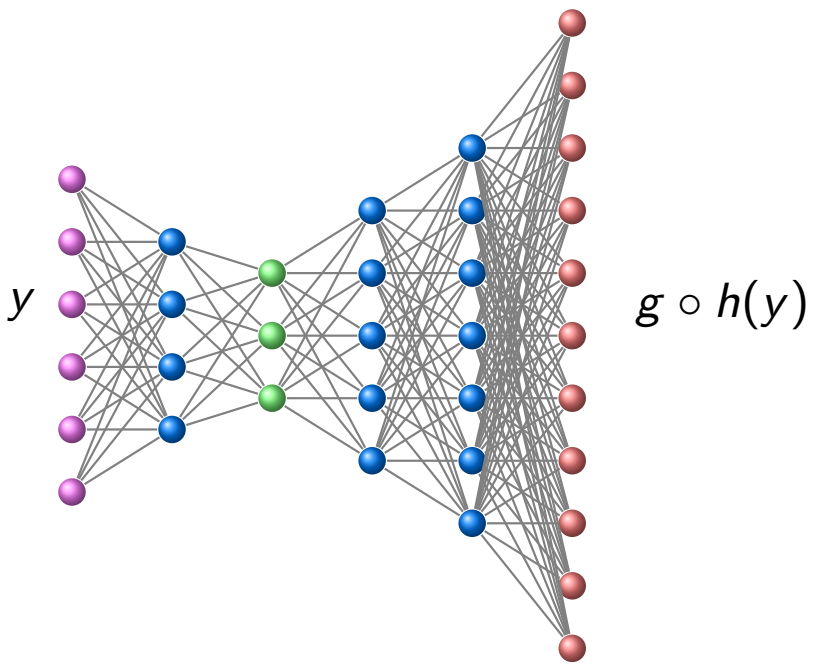
Nonlinear manifold approximation

If we know that $K = \{u(y) : y \in Y\}$ the image through a map u of a low-dimensional space Y, we can learn the map g from samples in Y by learning a compositional function

$$g \circ h$$

where $h : Y \to \mathbb{R}^m$. Given samples $y_1, \ldots, y_n$ in Y, this can be done by minimizing

$$\sum_{i=1}^{n} \|u(y_i) - g \circ h(y_i)\|_X^2$$



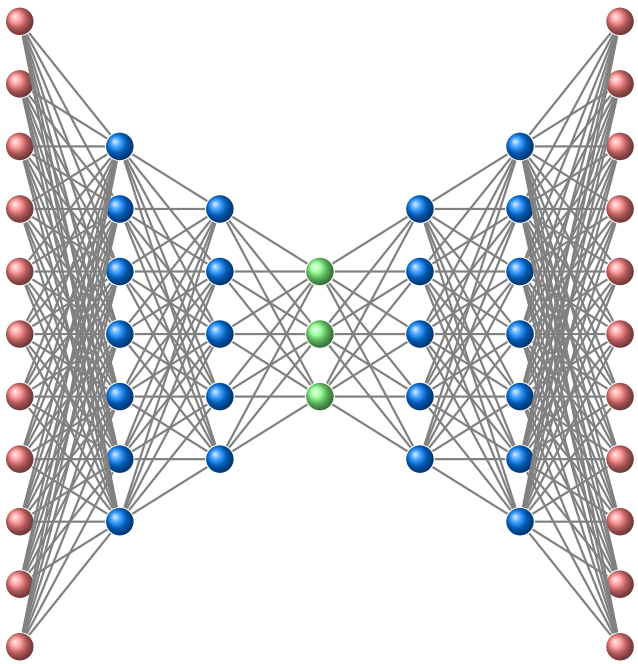
Nonlinear manifold approximation

If K in an infinite dimensional space X, a discretization is required.

A discretization can be represented by some encoder-decoder pair (E, R) with $E : X \to \mathbb{R}^N$ and $D : \mathbb{R}^N \to X$ (e.g. E could provide the values $E(u)$ of a function at the nodes of a mesh, and $D(E(u))$ a spline interpolation), and the functions g and h can be learned by solving

$$\min_{h,g} \sum_{i=1}^{n} \|u_i - D \circ g \circ h \circ E(u_i))\|_X^2$$

$$u \overset{E}{\mapsto} E(u)$$



$$g \circ h(E(u)) \overset{D}{\mapsto} D(g(h(E(u))))$$

The map $D \circ g \circ h \circ E$ is called an Operator Network that aims at approximating the identity map from K to X.

Nonlinear manifold approximation

For K a set of functions defined on a domain $\mathcal{X}$, with values in $\mathbb{R}$, an alternative is to consider

$$V_m = \{g(\cdot, a) : x \mapsto g(x, a) : a \in \mathbb{R}^m\}$$

with $g : \mathcal{X} \times \mathbb{R}^m \to \mathbb{R}$ in some high-dimensional approximation format (e.g. neural or tensor networks).

Function g can be learned (offline) from samples in K by solving

$$\min_{h,g} \sum_{i=1}^{n} \|u_i - g(\cdot, h(E(u_i))\|_X^2$$

where $E : K \to \mathbb{R}^N$ is some fixed discretization map (encoder) and $h : \mathbb{R}^N \to \mathbb{R}^m$. Here, no explicit decoder is used.

Outline

1. Manifold approximation
2. Linear approximation from point evaluations
3. Tensor networks approximation with point evaluations

Linear approximation from point evaluations

We consider the approximation of functions from a set

$$K \subset X \subset \mathbb{R}^{\mathcal{X}}$$

using point evaluations (standard information) and linear algorithms (linear approximation).

The best we can expect for the linear approximation of functions from a set K is characterized by sampling numbers $\rho_n(K)_X$ (for deterministic setting) or $\rho_n^{rand}(K)_X$ (for randomized setting) (see Part 1).

We assume that we are given a m-dimensional linear space V_m that is supposed to approximate well the set K.

The question is how to generate good points in $\mathcal{X}$ that allow to obtain an approximation in V_m with an error close to the best approximation error.

Interpolation

For a set of points $\boldsymbol{x} = (x_1, \ldots, x_m)$ unisolvent for V_m, we let $\mathcal{I}_{V_m} : X \to V_m$ be the corresponding interpolation (linear) operator.

We have

$$\|f - \mathcal{I}_{V_m} f\|_X \leq (1 + \|\mathcal{I}_{V_m}\|) \inf_{v \in V_m} \|f - v\|_X$$

For $(X, \|\cdot\|_\infty)$ the set of functions with bounded norm $\|f\|_\infty := \sup_{x \in \mathcal{X}} |f(x)|$, $\|\mathcal{I}_{V_m}\|$ is the Lebesgue constant, with

$$\|\mathcal{I}_{V_m}\| = \sup_{x \in \mathcal{X}} \sum_{i=1}^{m} |L_i(x)|$$

where $L_1, \ldots, L_m$ is the basis of V_m satisfying the interpolation property ($L_i(x_j) = \delta_{ij}$ for all i, j).

Interpolation

For a set of points $\boldsymbol{x} = (x_1, \ldots, x_m)$ unisolvent for V_m, we let $\mathcal{I}_{V_m} : X \to V_m$ be the corresponding interpolation (linear) operator.

We have

$$\|f - \mathcal{I}_{V_m} f\|_X \leq (1 + \|\mathcal{I}_{V_m}\|) \inf_{v \in V_m} \|f - v\|_X$$

For $(X, \|\cdot\|_\infty)$ the set of functions with bounded norm $\|f\|_\infty := \sup_{x \in \mathcal{X}} |f(x)|$, $\|\mathcal{I}_{V_m}\|$ is the Lebesgue constant, with

$$\|\mathcal{I}_{V_m}\| = \sup_{x \in \mathcal{X}} \sum_{i=1}^{m} |L_i(x)|$$

where $L_1, \ldots, L_m$ is the basis of V_m satisfying the interpolation property ($L_i(x_j) = \delta_{ij}$ for all i, j).

For univariate functions and classical spaces V_m (polynomials, splines), the theory is well established and suitable choices of points are available.

Except in very specific cases (e.g. piecewise constant or linear approximation), $\|\mathcal{I}_{V_m}\|$ grows with m. The question is to find good points such that $\|\mathcal{I}_{V_m}\|$ grows not too fast with m.

Empirical interpolation

Given a space V_m with basis $\varphi_1, \ldots, \varphi_m$, a general greedy algorithm has been proposed in [Maday et al 2009] to construct interpolation points, called magic points.

The idea is to construct a good sequence of spaces $W_k = span\{\psi_1, \ldots, \psi_k\}$ for the approximation of the discrete set $\{\varphi_i : 1 \leq i \leq m\}$ in $(X, \|\cdot\|_\infty)$, and associated interpolation points.

Empirical interpolation

Given a space V_m with basis $\varphi_1, \ldots, \varphi_m$, a general greedy algorithm has been proposed in [Maday et al 2009] to construct interpolation points, called magic points.

The idea is to construct a good sequence of spaces $W_k = span\{\psi_1, \ldots, \psi_k\}$ for the approximation of the discrete set $\{\varphi_i : 1 \leq i \leq m\}$ in $(X, \|\cdot\|_\infty)$, and associated interpolation points.

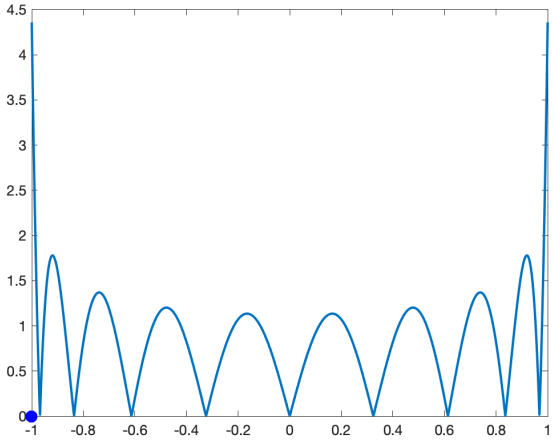
Starting from $V_0 = \{0\}$, we define

$$i_k \in \arg\max_{1 \leq i \leq m} \|\varphi_i - \mathcal{I}_{W_{k-1}} \varphi_i\|_\infty, \quad \psi_k = \varphi_{i_k} - \mathcal{I}_{W_{k-1}} \varphi_{i_k}$$

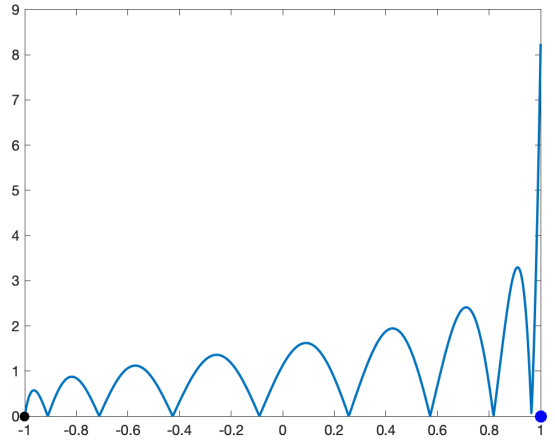
where $\mathcal{I}_{W_{k-1}}$ is the interpolation onto W_{k-1} using points $(x_1, \ldots x_{k-1})$, and define

$$x_k \in \arg\max_{x \in \mathcal{X}} |\psi_k(x)|.$$

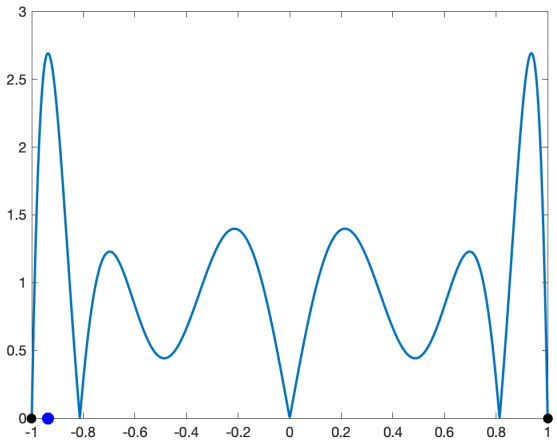
Empirical interpolation



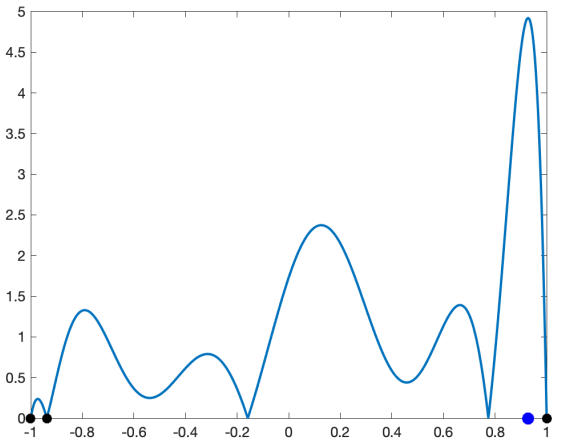
(a) $k = 1$



(b) $k = 2$



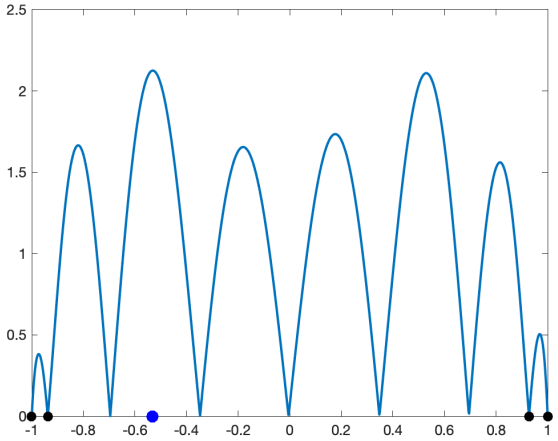
(c) $k = 3$



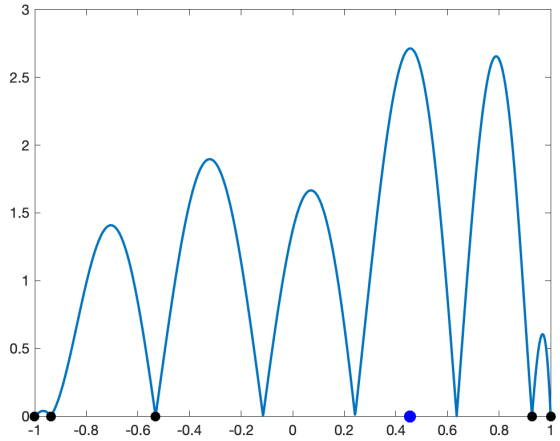
(d) $k = 4$

Figure: Polynomial space $V_m = \mathbb{P}_9$ on $[-1, 1]$. Function $|\psi_k(x)|$ and corresponding interpolation point $x_k = \arg\max_x |\psi_k(x)|$

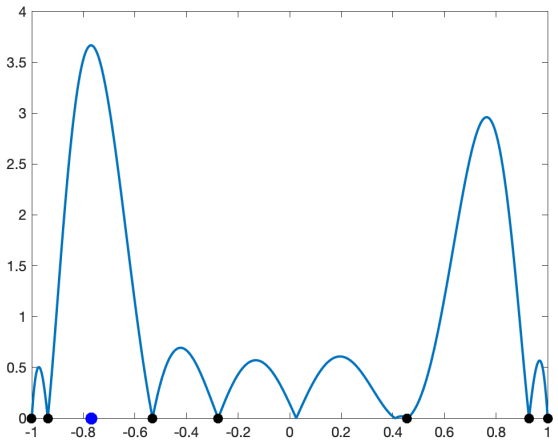
Empirical interpolation



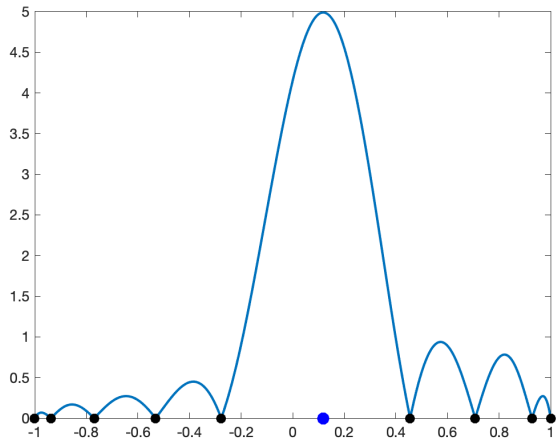
(a) $k = 5$



(b) $k = 6$



(c) $k = 8$



(d) $k = 10$

Figure: Polynomial space $V_m = \mathbb{P}_9$ on $[-1, 1]$. Function $|\psi_k(x)|$ and corresponding interpolation point $x_k = \arg\max_x |\psi_k(x)|$

Empirical interpolation

In the context of adaptive approximation in a sequence of spaces $V_1 \subset \ldots \subset V_m \subset \ldots$, and in order to recycle interpolation points, we modify the algorithm by simply taking $W_k = V_k$.

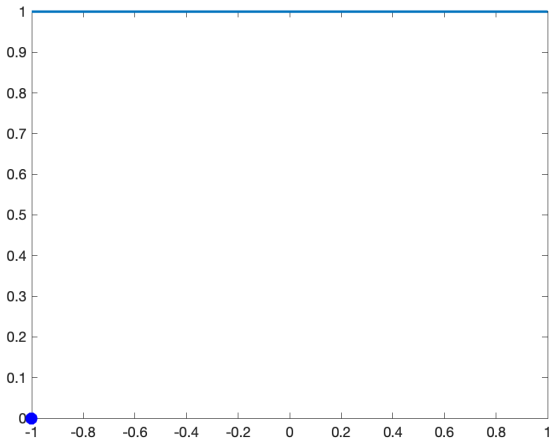
Letting $V_0 = \{0\}$, we define

$$\psi_k = \varphi_k - \mathcal{I}_{V_{k-1}} \varphi_k$$

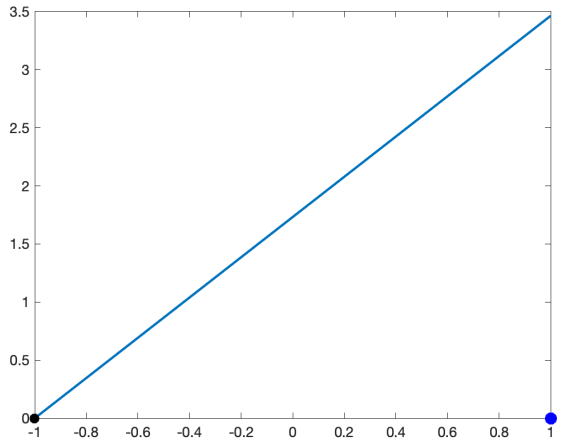
where $\mathcal{I}_{V_{k-1}}$ is the interpolation onto V_{k-1} using points $(x_1, \ldots x_{k-1})$, and define

$$x_k \in \arg\max_{x \in \mathcal{X}} |\psi_k(x)|.$$

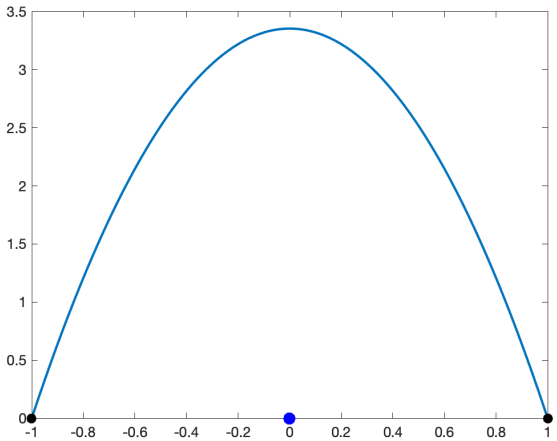
Empirical interpolation — adaptive setting



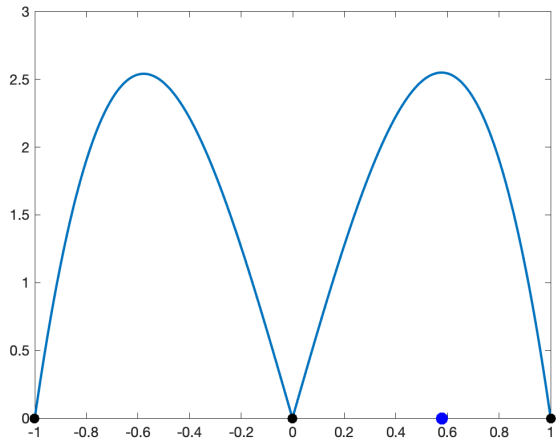
(a) $k = 1$



(b) $k = 2$



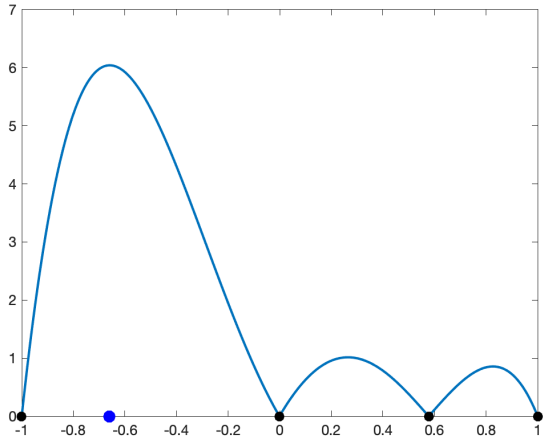
(c) $k = 3$



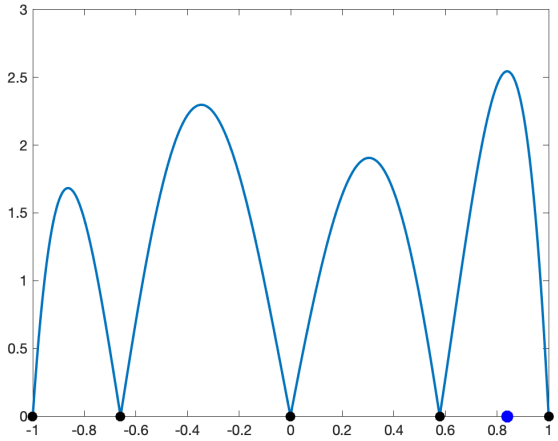
(d) $k = 4$

Figure: Polynomial space $V_m = \mathbb{P}_9$ on $[-1, 1]$. Function $|\psi_k(x)|$ and corresponding interpolation point $x_k = \arg\max_x |\psi_k(x)|$

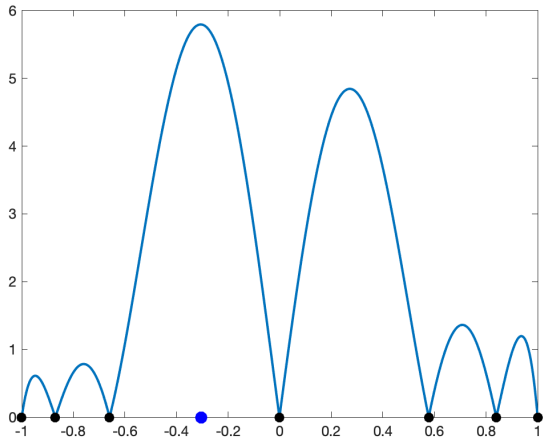
Empirical interpolation — adaptive setting



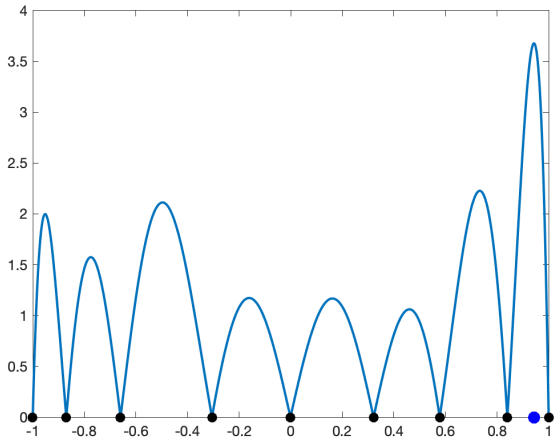
(a) $k = 5$



(b) $k = 6$



(c) $k = 8$



(d) $k = 10$

Figure: Polynomial space $V_m = \mathbb{P}_9$ on $[-1, 1]$. Function $|\psi_k(x)|$ and corresponding interpolation point $x_k = \arg\max_x |\psi_k(x)|$

Empirical interpolation based on feature map

Another strategy can be defined as follows. Let $\varphi(x) = (\varphi_1(x), \ldots, \varphi_m(x)) \in \mathbb{R}^m$, where $\varphi : \mathcal{X} \to \mathbb{R}^m$ is the feature map associated with V_m. The feature space $\mathbb{R}^m$ is equipped with the Euclidian norm $\|\cdot\|$.

The idea is to construct an increasing sequence of spaces

$$U_k = span\{\varphi(x_1), \ldots, \varphi(x_k)\} \subset \mathbb{R}^m$$

for the approximation of the manifold $\{\varphi(x) : x \in \mathcal{X}\}$.

Empirical interpolation based on feature map

Another strategy can be defined as follows. Let $\boldsymbol{\varphi}(x) = (\varphi_1(x), \ldots, \varphi_m(x)) \in \mathbb{R}^m$, where $\boldsymbol{\varphi} : \mathcal{X} \to \mathbb{R}^m$ is the feature map associated with V_m. The feature space $\mathbb{R}^m$ is equipped with the Euclidian norm $\|\cdot\|$.

The idea is to construct an increasing sequence of spaces

$$U_k = span\{\boldsymbol{\varphi}(x_1), \ldots, \boldsymbol{\varphi}(x_k)\} \subset \mathbb{R}^m$$

for the approximation of the manifold $\{\boldsymbol{\varphi}(x) : x \in \mathcal{X}\}$.

Starting from $U_0 = \{0\}$, we define

$$x_k \in \arg\max_{x \in \mathcal{X}} \Lambda_k(x), \quad \Lambda_k(x) = \|\boldsymbol{\varphi}(x) - P_{U_{k-1}} \boldsymbol{\varphi}(x)\|_2^2$$

where $P_{U_{k-1}}$ is the orthogonal projection from $\mathbb{R}^m$ to U_{k-1}.

Empirical interpolation based on feature map

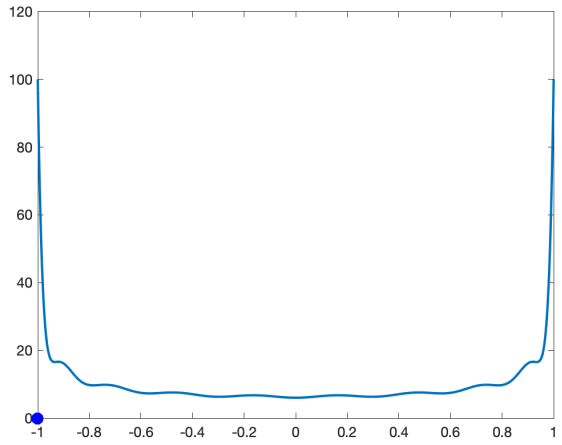
Let $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m)$ be the orthonormal basis of $\mathbb{R}^m$ defined by

$$\boldsymbol{e}_k \propto \boldsymbol{\varphi}(x_k) - P_{U_{k-1}} \boldsymbol{\varphi}(x_k), \quad \|\boldsymbol{e}_k\|_2 = 1.$$

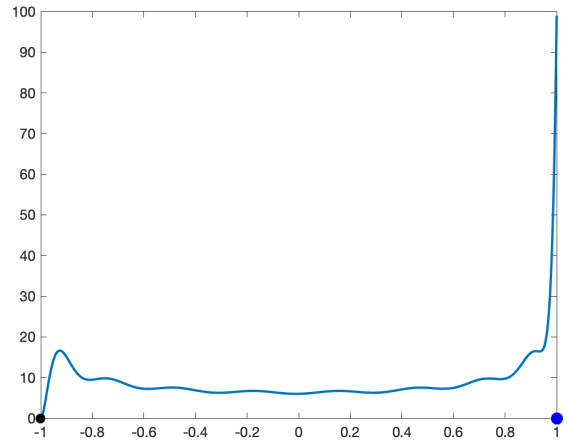
If V_m is a Hilbert space and the functions φ_i form an orthonormal basis of V_m, then the functions $\psi_i(x) = \boldsymbol{\varphi}(x)^T \boldsymbol{e}_i$ also form an orthonormal basis of V_m and

$$\Lambda_k(x) = \sum_{i=k}^{m} \psi_i(x)^2 = \|\boldsymbol{\varphi}(x)\|_2^2 - \sum_{i=1}^{k-1} \psi_i(x)^2$$

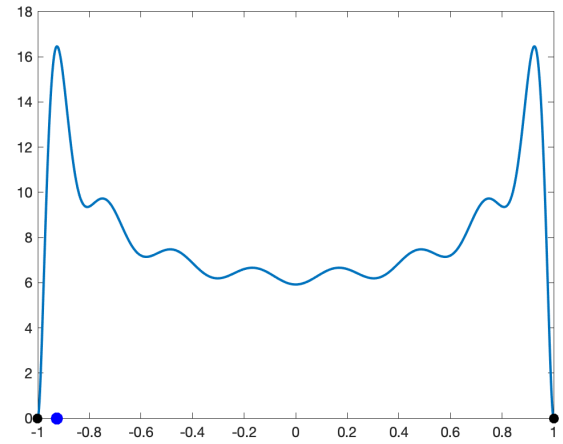
Empirical interpolation based on feature map



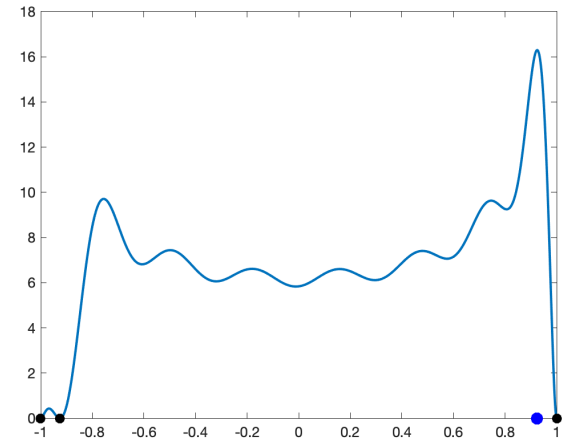
(a) $k = 1$



(b) $k = 2$



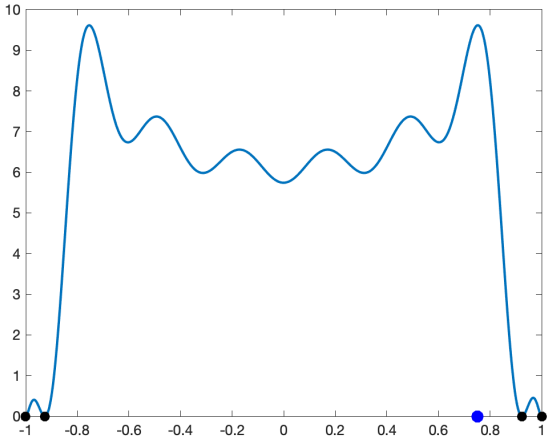
(c) $k = 3$



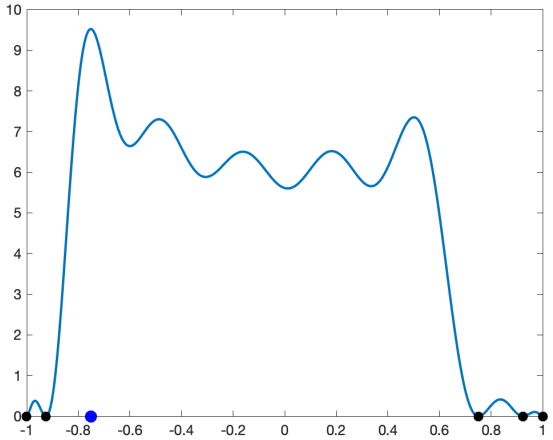
(d) $k = 4$

Figure: Polynomial space $V_m = \mathbb{P}_9$ on $[-1, 1]$. Function $\Lambda_k(x)$ and corresponding interpolation point $x_k = \arg\max_x \Lambda_k(x)$

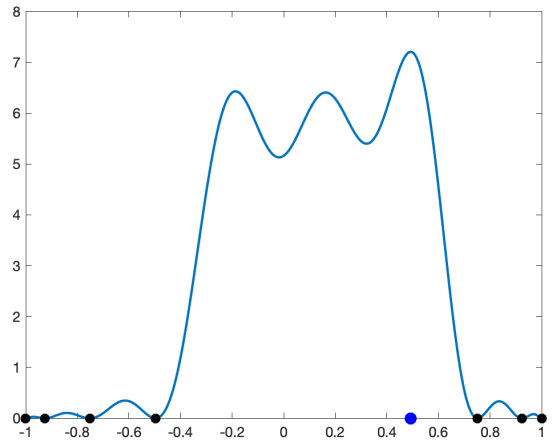
Empirical interpolation based on feature map



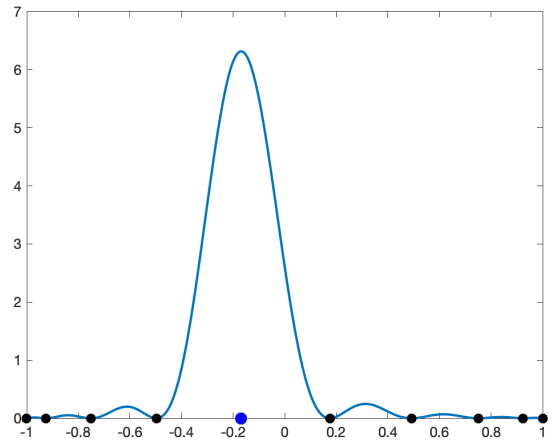
(a) $k = 5$



(b) $k = 6$



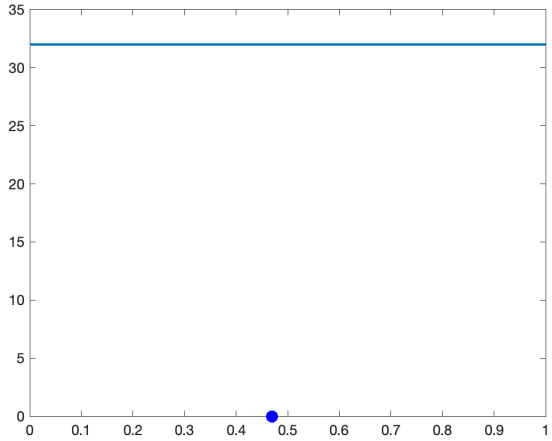
(c) $k = 8$



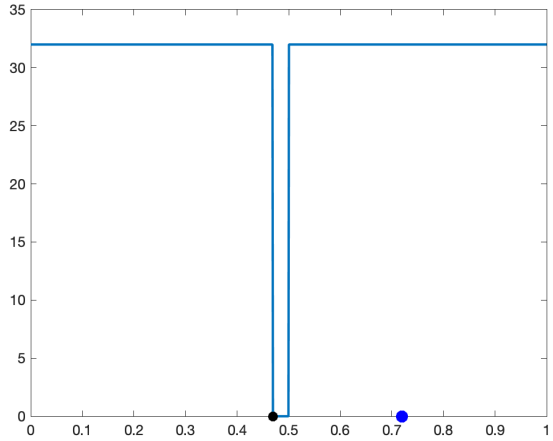
(d) $k = 10$

Figure: Polynomial space $V_m = \mathbb{P}_9$ on $[-1, 1]$. Function $\Lambda_k(x)$ and corresponding interpolation point $x_k = \arg\max_x \Lambda_k(x)$

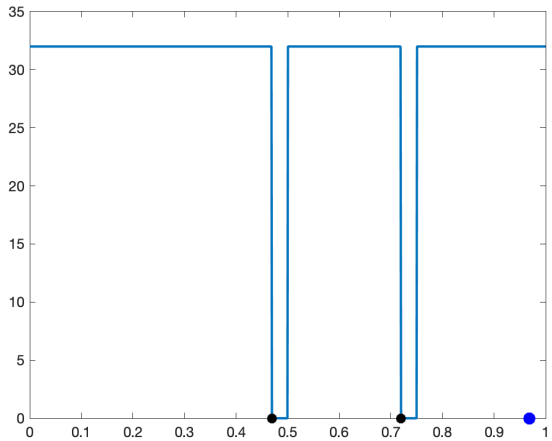
Empirical interpolation based on feature map



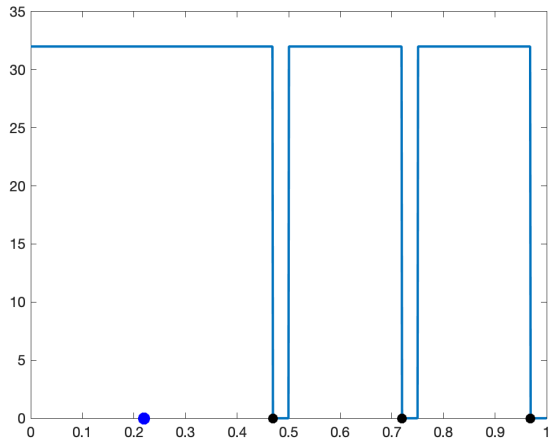
(a) $k = 1$



(b) $k = 2$



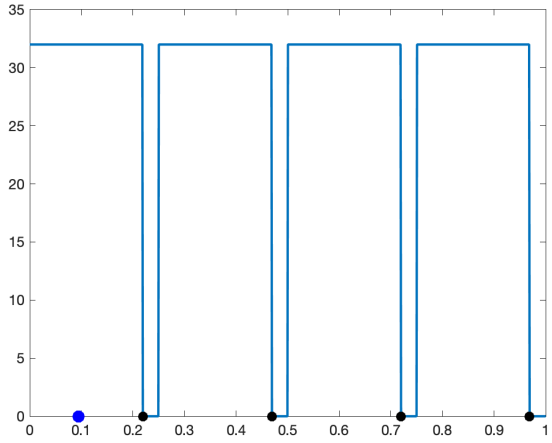
(c) $k = 3$



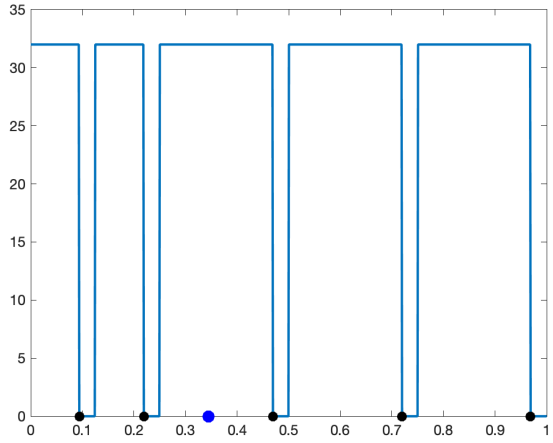
(d) $k = 4$

Figure: Haar wavelets space V_m on $[0, 1]$, with resolution 5. Function $\Lambda_k(x)$ and corresponding interpolation point $x_k = \arg\max_x \Lambda_k(x)$.

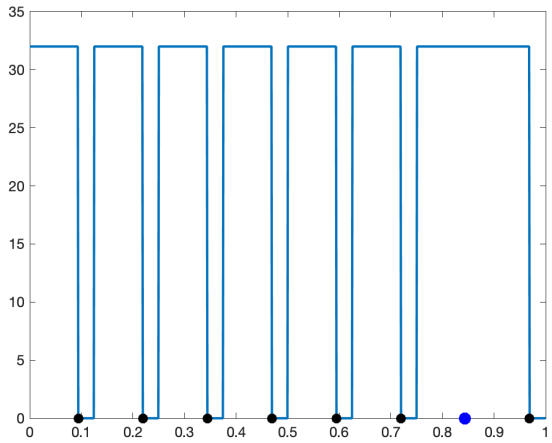
Empirical interpolation based on feature map



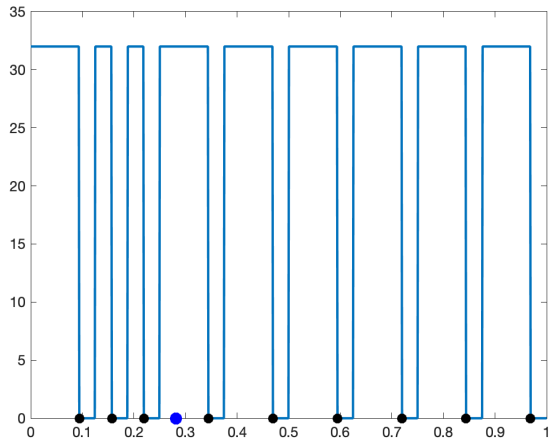
(a) $k = 5$



(b) $k = 6$



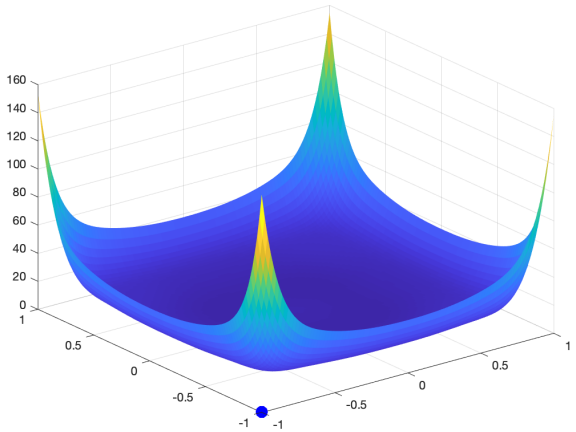
(c) $k = 8$



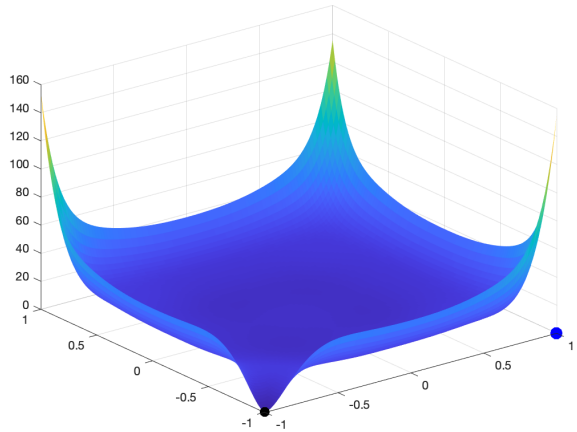
(d) $k = 10$

Figure: Haar wavelets space V_m on $[0, 1]$, with resolution 5. Function $\Lambda_k(x)$ and corresponding interpolation point $x_k = \arg\max_x \Lambda_k(x)$.

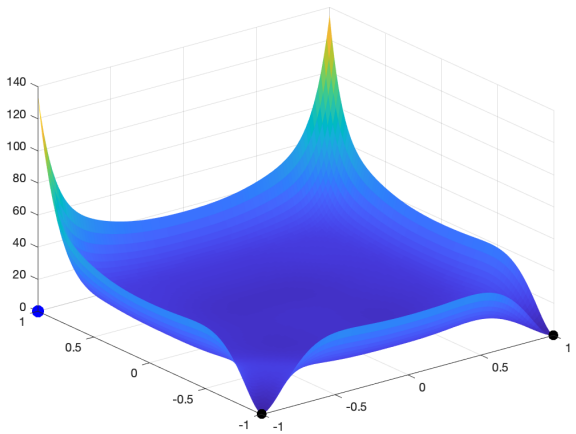
Empirical interpolation based on feature map



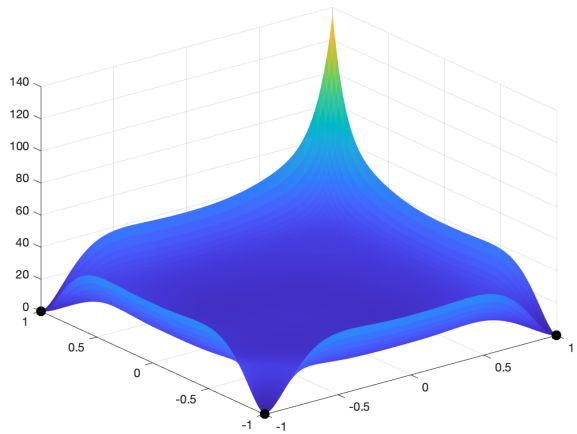
(a) $k = 1$



(b) $k = 2$



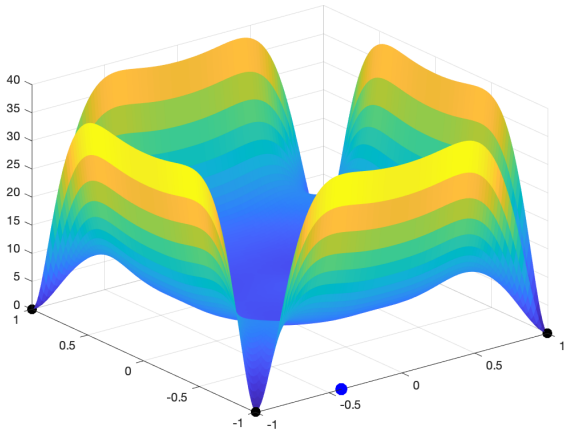
(c) $k = 3$



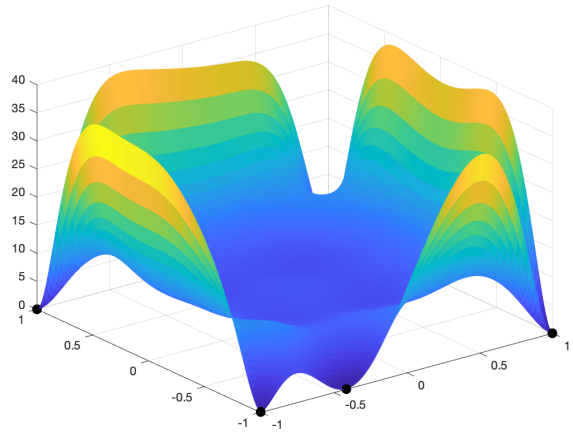
(d) $k = 4$

Figure: Bivariate polynomial space $V_m = \mathbb{P}_4$ on $[-1, 1]^2$. Function $\Lambda_k(x)$ and corresponding interpolation point $x_k = \arg\max_x \Lambda_k(x)$.

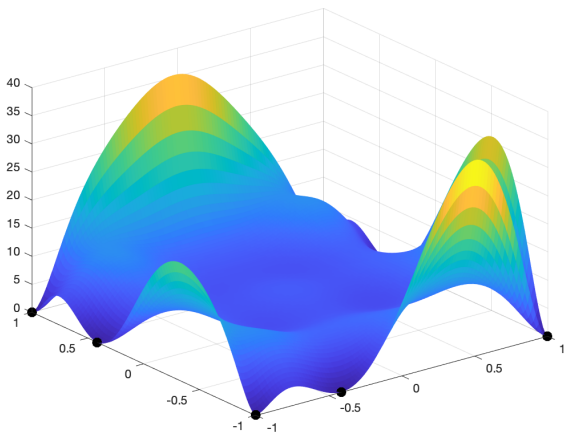
Empirical interpolation based on feature map



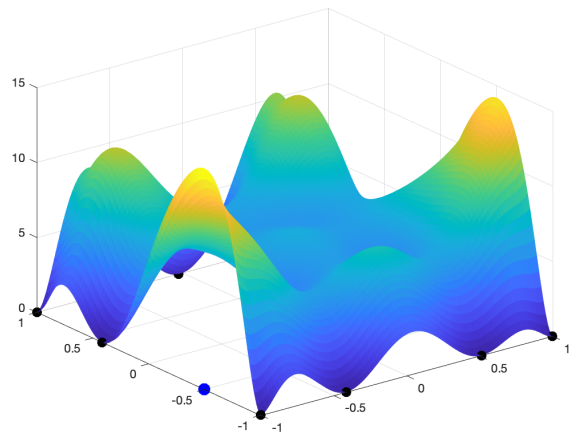
(a) $k = 5$



(b) $k = 6$



(c) $k = 8$



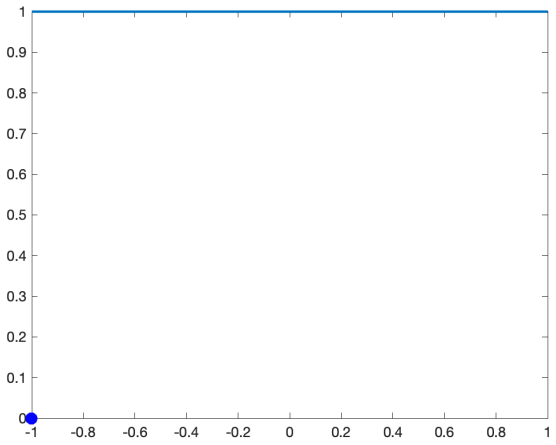
(d) $k = 10$

Figure: Bivariate polynomial space $V_m = \mathbb{P}_4$ on $[-1, 1]^2$. Function $\Lambda_k(x)$ and corresponding interpolation point $x_k = \arg\max_x \Lambda_k(x)$.

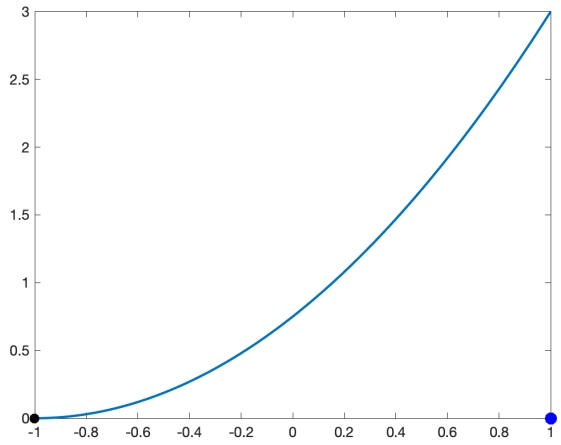
Empirical interpolation based on feature map — adaptive setting

In the context of adaptive approximation in a sequence of spaces $V_1 \subset \ldots \subset V_m \subset \ldots$, and in order to recycle interpolation points, we modify the algorithm by considering at step k the feature map φ associated with the basis of V_k.

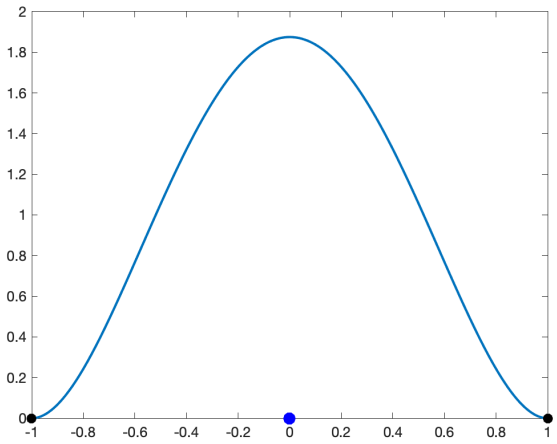
Empirical interpolation based on feature map — adaptive setting



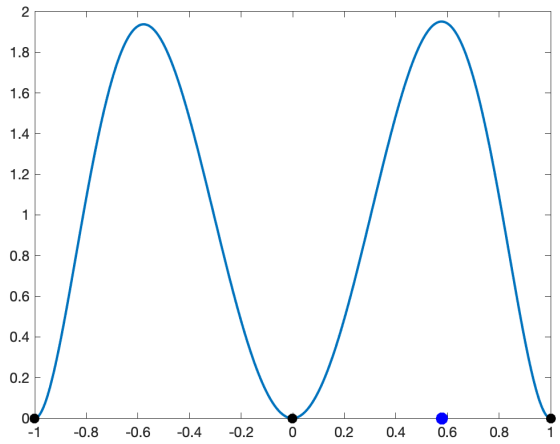
(a) $k = 1$



(b) $k = 2$



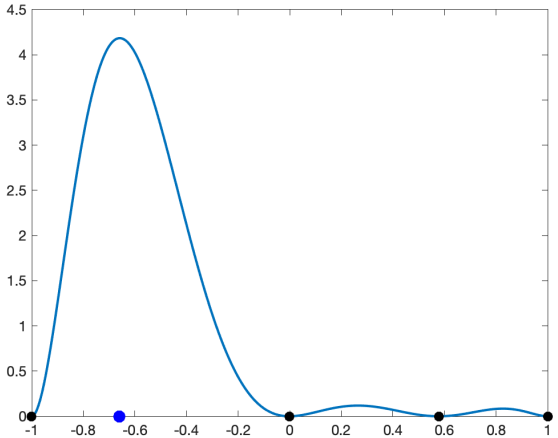
(c) $k = 3$



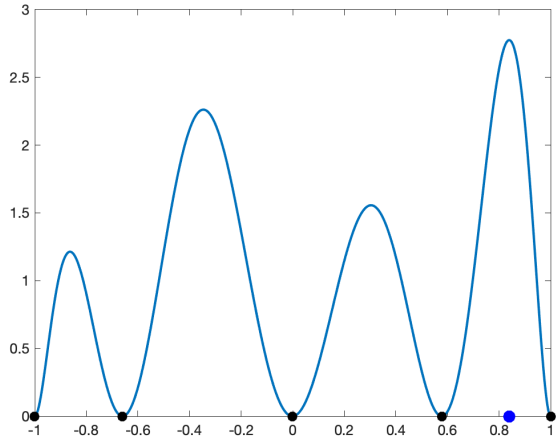
(d) $k = 4$

Figure: Polynomial space $V_m = \mathbb{P}_9$ on [-1,1]. Function $\Lambda_k(x)$ and corresponding interpolation point $x_k = \arg\max_x \Lambda_k(x)$.

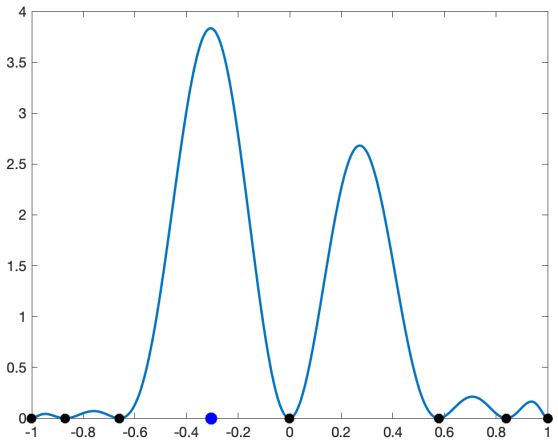
Empirical interpolation based on feature map — adaptive setting



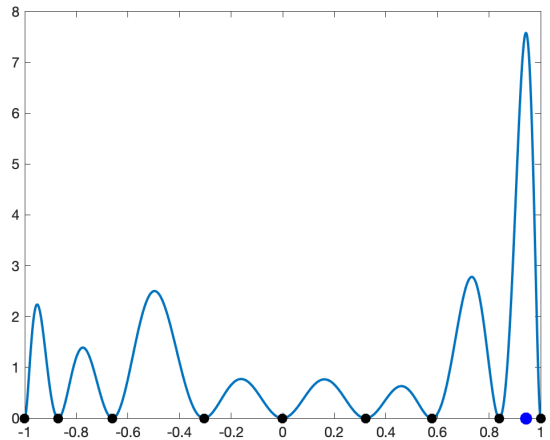
(a) $k = 5$



(b) $k = 6$



(c) $k = 8$



(d) $k = 10$

Figure: Polynomial space $V_m = \mathbb{P}_9$ on [-1,1]. Function $\Lambda_k(x)$ and corresponding interpolation point $x_k = \arg\max_x \Lambda_k(x)$.

Interpolation in RKHS

A reproducing kernel Hilbert space (RKHS) H is a Hilbert space of functions defined on $\mathcal{X}$ such that the point evaluation $\delta_x : f : x \mapsto f(x)$ is a continuous linear map. There is a so called reproducing kernel k such that $k(x, \cdot)$ is the Riesz representer of δ_x, that is

$$f(x) = (f, k(x, \cdot))_H,$$

where $(\cdot, \cdot)_H$ is the inner product on H.

Interpolation in RKHS

A reproducing kernel Hilbert space (RKHS) H is a Hilbert space of functions defined on $\mathcal{X}$ such that the point evaluation $\delta_x : f : x \mapsto f(x)$ is a continuous linear map. There is a so called reproducing kernel k such that $k(x, \cdot)$ is the Riesz representer of δ_x, that is

$$f(x) = (f, k(x, \cdot))_H,$$

where $(\cdot, \cdot)_H$ is the inner product on H.

For given points $\boldsymbol{x} = (x_1, \ldots, x_k)$, the interpolation operator $\mathcal{I}_{W_k}$ onto the space $W_k = \mathit{span}\{k(\cdot, x_1), \ldots, k(\cdot, x_k)\}$ is defined by

$$\mathcal{I}_{W_k} f(x) = k(x, \boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{x})^{-1} f(\boldsymbol{x})$$

where $k(\boldsymbol{x}, \boldsymbol{y}) = (k(x_i, y_j))_{i,j}$ and $f(\boldsymbol{x}) = (f(x_j))_j$.

Interpolation in RKHS

A reproducing kernel Hilbert space (RKHS) H is a Hilbert space of functions defined on $\mathcal{X}$ such that the point evaluation $\delta_x : f : x \mapsto f(x)$ is a continuous linear map. There is a so called reproducing kernel k such that $k(x, \cdot)$ is the Riesz representer of δ_x, that is

$$f(x) = (f, k(x, \cdot))_H,$$

where $(\cdot, \cdot)_H$ is the inner product on H.

For given points $\boldsymbol{x} = (x_1, \ldots, x_k)$, the interpolation operator $\mathcal{I}_{W_k}$ onto the space $W_k = span\{k(\cdot, x_1), \ldots, k(\cdot, x_k)\}$ is defined by

$$\mathcal{I}_{W_k} f(x) = k(x, \boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{x})^{-1} f(\boldsymbol{x})$$

where $k(\boldsymbol{x}, \boldsymbol{y}) = (k(x_i, y_j))_{i,j}$ and $f(\boldsymbol{x}) = (f(x_j))_j$. The operator $\mathcal{I}_{W_k}$ is the

H-orthogonal projection onto W_k, which provides the element of best approximation of a function in W_k. Indeed, for $f \in H$, the interpolation conditions

$$\mathcal{I}_{W_k} f(x_i) = f(x_i), \quad 1 \leq i \leq k,$$

are equivalent to

$$(k(\cdot, x_i), \mathcal{I}_{W_k} f - f)_H = 0, \quad 1 \leq i \leq k,$$

that is $\mathcal{I}_{W_k} f - f$ is orthogonal to W_k.

Interpolation in RKHS

The error of interpolation at point $x \in \mathcal{X}$ is such that

$$\begin{aligned} |f(x) - \mathcal{I}_{W_k} f(x)| &= |\left(k(x,\cdot), \mathcal{I}_{W_k} f - f\right)_H| \\ &= |\left(k(x,\cdot) - \mathcal{I}_{W_k} k(x,\cdot), \mathcal{I}_{W_k} f - f\right)_H| \\ &\leq \|k(x,\cdot) - \mathcal{I}_{W_k} k(x,\cdot)\|_H \|f\|_H \end{aligned}$$

Interpolation in RKHS

The error of interpolation at point $x \in \mathcal{X}$ is such that

$$\begin{aligned} |f(x) - \mathcal{I}_{W_k} f(x)| &= |\left(k(x,\cdot), \mathcal{I}_{W_k} f - f\right)_H| \\ &= |\left(k(x,\cdot) - \mathcal{I}_{W_k} k(x,\cdot), \mathcal{I}_{W_k} f - f\right)_H| \\ &\leq \|k(x,\cdot) - \mathcal{I}_{W_k} k(x,\cdot)\|_H \|f\|_H \end{aligned}$$

A natural definition of a new basis function $k(x_{k+1}, \cdot)$ is to consider a point x_{k+1} where the error bound is maximum, that is

$$x_{k+1} \in \arg\max_{x \in \mathcal{X}} \Lambda_k(x),$$

with

$$\Lambda_k(x) = \|k(x,\cdot) - \mathcal{I}_{W_k} k(x,\cdot)\|_H^2 = k(x,x) - k(x,\boldsymbol{x}) k(\boldsymbol{x},\boldsymbol{x})^{-1} k(\boldsymbol{x},x).$$

Interpolation in RKHS

A finite dimensional space V_m with basis $\varphi_1, \ldots, \varphi_m$ defines a RKHS with kernel

$$k(x, y) = \boldsymbol{\varphi}(x)^T \boldsymbol{\varphi}(y), \quad \boldsymbol{\varphi}(x) := (\varphi_1(x), \ldots, \varphi_m(x))$$

A sequential interpolation method consists in defining a sequence of points $(x_k)_{k \geq 1}$ and corresponding spaces $W_k = span\{k(x_1, \cdot), \ldots, k(x_k, \cdot)\}$ such that

$$x_{k+1} = \arg\max_{x \in \mathcal{X}} \Lambda_k(x),$$

where

$$\Lambda_k(x) = \|\boldsymbol{\varphi}(x)\|_2^2 - \boldsymbol{\varphi}(x)^T \boldsymbol{\varphi}(\boldsymbol{x})(\boldsymbol{\varphi}(\boldsymbol{x})\boldsymbol{\varphi}(\boldsymbol{x})^T)^{-1} \boldsymbol{\varphi}(\boldsymbol{x})^T \boldsymbol{\varphi}(x)$$

with $\boldsymbol{x} = (x_1, \ldots, x_k)$ and $\boldsymbol{\varphi}(\boldsymbol{x}) = (\varphi_i(x_j))_{1 \leq i,j \leq k}$.

Interpolation in RKHS

A finite dimensional space V_m with basis $\varphi_1, \dots, \varphi_m$ defines a RKHS with kernel

$$k(x, y) = \boldsymbol{\varphi}(x)^T \boldsymbol{\varphi}(y), \quad \boldsymbol{\varphi}(x) := (\varphi_1(x), \dots, \varphi_m(x))$$

A sequential interpolation method consists in defining a sequence of points $(x_k)_{k \geq 1}$ and corresponding spaces $W_k = span\{k(x_1, \cdot), \dots, k(x_k, \cdot)\}$ such that

$$x_{k+1} = \arg\max_{x \in \mathcal{X}} \Lambda_k(x),$$

where

$$\Lambda_k(x) = \|\boldsymbol{\varphi}(x)\|_2^2 - \boldsymbol{\varphi}(x)^T \boldsymbol{\varphi}(\boldsymbol{x})(\boldsymbol{\varphi}(\boldsymbol{x})\boldsymbol{\varphi}(\boldsymbol{x})^T)^{-1} \boldsymbol{\varphi}(\boldsymbol{x})^T \boldsymbol{\varphi}(x)$$

with $\boldsymbol{x} = (x_1, \dots, x_k)$ and $\boldsymbol{\varphi}(\boldsymbol{x}) = (\varphi_i(x_j))_{1 \leq i,j \leq k}$.

In bayesian regression with gaussian processes (with noisy-free observations), the function $\Lambda_k(x)$ is the variance of the conditional gaussian process given observations at points $\boldsymbol{x} = (x_1, \dots, x_k)$.

Note that the obtained sequence of points only depends on the space V_m.

Interpolation in RKHS

A finite dimensional space V_m with basis $\varphi_1, \dots, \varphi_m$ defines a RKHS with kernel

$$k(x, y) = \boldsymbol{\varphi}(x)^T \boldsymbol{\varphi}(y), \quad \boldsymbol{\varphi}(x) := (\varphi_1(x), \dots, \varphi_m(x))$$

A sequential interpolation method consists in defining a sequence of points $(x_k)_{k \geq 1}$ and corresponding spaces $W_k = span\{k(x_1, \cdot), \dots, k(x_k, \cdot)\}$ such that

$$x_{k+1} = \arg\max_{x \in \mathcal{X}} \Lambda_k(x),$$

where

$$\Lambda_k(x) = \|\boldsymbol{\varphi}(x)\|_2^2 - \boldsymbol{\varphi}(x)^T \boldsymbol{\varphi}(\boldsymbol{x})(\boldsymbol{\varphi}(\boldsymbol{x})\boldsymbol{\varphi}(\boldsymbol{x})^T)^{-1} \boldsymbol{\varphi}(\boldsymbol{x})^T \boldsymbol{\varphi}(x)$$

with $\boldsymbol{x} = (x_1, \dots, x_k)$ and $\boldsymbol{\varphi}(\boldsymbol{x}) = (\varphi_i(x_j))_{1 \leq i,j \leq k}$.

In bayesian regression with gaussian processes (with noisy-free observations), the function $\Lambda_k(x)$ is the variance of the conditional gaussian process given observations at points $\boldsymbol{x} = (x_1, \dots, x_k)$.

Note that the obtained sequence of points only depends on the space V_m.

Letting $U_k = span\{\boldsymbol{\varphi}(x_1), \dots, \boldsymbol{\varphi}(x_k)\} \subset \mathbb{R}^m$, we note that

$$\Lambda_k(x) = \|\boldsymbol{\varphi}(x) - P_{U_{k-1}} \boldsymbol{\varphi}(x)\|_2^2$$

This is equivalent to the previously presented empirical interpolation based on feature map.

Least squares approximation

Consider the approximation of a function f in $X = L^2_\mu(\mathcal{X})$ equipped with the norm

$$\|f\|^2 = \int f(x)^2 d\mu(x).$$

Given a m-dimensional space V_m in $L^2_\mu(\mathcal{X})$, a weighted least-squares approximation $\hat{f}_m \in V_m$ is defined by minimizing

$$\frac{1}{n} \sum_{i=1}^{n} w_i (v(x_i) - f(x_i))^2$$

over $v \in V_m$, for some suitably chosen points $\boldsymbol{x} = (x_1, \ldots, x_n)$ and corresponding weights $\boldsymbol{w} = (w_1, \ldots, w_n)$.

Least squares approximation

This is equivalent to minimize

$$\|f - v\|_n^2$$

where $\|\cdot\|_n^2$ is a semi-norm defined by

$$\|f\|_n^2 = \frac{1}{n}\sum_{i=1}^{n} w_i f(x_i)^2$$

Least squares approximation

This is equivalent to minimize

$$\|f - v\|_n^2$$

where $\|\cdot\|_n^2$ is a semi-norm defined by

$$\|f\|_n^2 = \frac{1}{n}\sum_{i=1}^{n} w_i f(x_i)^2$$

Assuming that the x_i are i.i.d. samples from a distribution ν defined by

$$d\nu(x) = w(x)^{-1} d\mu(x),$$

and the weights $w_i = w(x_i)$, then for all $f \in L^2_\mu$

$$\mathbb{E}(\|f\|_n^2) = \mathbb{E}_{x\sim\nu}(w(x)f(x)^2) = \mathbb{E}_{x\sim\mu}(f(x)^2) = \|f\|^2$$

Least squares approximation

Given an L^2_μ-orthonormal basis $\varphi_1(x), ..., \varphi_m(x)$ of V_m, and letting $\boldsymbol{\varphi}(x) = (\varphi_1(x), ..., \varphi_m(x))^T \in \mathbb{R}^m$, a function $v \in V_m$ can be written

$$v(x) = \sum_{i=1}^{m} a_i \varphi_i(x) = \boldsymbol{\varphi}(x)^T \boldsymbol{a}$$

We have

$$\|v\|^2 = \|\boldsymbol{a}\|_2^2$$

and

$$\|v\|_n^2 = \boldsymbol{a}^T \boldsymbol{G} \boldsymbol{a}$$

where $\boldsymbol{G}$ is the empirical Gram matrix (or weighted information matrix) given by

$$\boldsymbol{G} := \boldsymbol{G}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} w_i \boldsymbol{\varphi}(x_i) \boldsymbol{\varphi}(x_i)^T.$$

Least squares approximation

Given an L^2_μ-orthonormal basis $\varphi_1(x), ..., \varphi_m(x)$ of V_m, and letting $\boldsymbol{\varphi}(x) = (\varphi_1(x), ..., \varphi_m(x))^T \in \mathbb{R}^m$, a function $v \in V_m$ can be written

$$v(x) = \sum_{i=1}^{m} a_i \varphi_i(x) = \boldsymbol{\varphi}(x)^T \boldsymbol{a}$$

We have

$$\|v\|^2 = \|\boldsymbol{a}\|_2^2$$

and

$$\|v\|_n^2 = \boldsymbol{a}^T \boldsymbol{G} \boldsymbol{a}$$

where $\boldsymbol{G}$ is the empirical Gram matrix (or weighted information matrix) given by

$$\boldsymbol{G} := \boldsymbol{G}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} w_i \boldsymbol{\varphi}(x_i) \boldsymbol{\varphi}(x_i)^T.$$

We have

$$\lambda_{min}(\boldsymbol{G}) \|v\|^2 \leq \|v\|_n^2 \leq \lambda_{max}(\boldsymbol{G}) \|v\|^2 \quad \forall v \in V_m.$$

The quality of least-squares projection is related to how much $\boldsymbol{G}$ deviates from the identity.

Optimal design of experiments

Consider the model

$$Y = f(X) + \epsilon$$

where $X \sim \mu$ and $\epsilon \sim \mathcal{N}(0, \lambda)$ is independent of X, that corresponds to noisy evaluations of a function f.

For given points $\boldsymbol{x} = (x_1, \ldots, x_n)$ we have access to $\boldsymbol{y} = (y_1, \ldots, y_n)$ such that

$$y_i = f(x_i) + \epsilon_i$$

with $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n) \sim \mathcal{N}(0, \Lambda)$ independent of $\boldsymbol{x}$.

Optimal design of experiments

Consider the model

$$Y = f(X) + \epsilon$$

where $X \sim \mu$ and $\epsilon \sim \mathcal{N}(0, \lambda)$ is independent of X, that corresponds to noisy evaluations of a function f.

For given points $\boldsymbol{x} = (x_1, \ldots, x_n)$ we have access to $\boldsymbol{y} = (y_1, \ldots, y_n)$ such that

$$y_i = f(x_i) + \epsilon_i$$

with $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n) \sim \mathcal{N}(0, \Lambda)$ independent of $\boldsymbol{x}$.

A weighted least-squares estimate $\hat{f}_m$ is then obtained by solving

$$\min_{v \in V_m} \frac{1}{n} \sum_{i=1}^{n} w_i (v(x_i) - y_i)^2$$

Letting $\Phi := \Phi(\boldsymbol{x}) = (\varphi_j(x_i))_{1 \leq i \leq n, 1 \leq j \leq m}$ (the design matrix) and $\boldsymbol{W} = diag(\boldsymbol{w})$ the weight matrix, we have

$$\hat{f}_m(x) = \boldsymbol{\varphi}(x)^T \hat{\boldsymbol{a}}, \quad \hat{\boldsymbol{a}} = \boldsymbol{G}^{-1} \Phi^T \boldsymbol{W} \boldsymbol{y}$$

with

$$\boldsymbol{G} := \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{w}) = \Phi^T \boldsymbol{W} \Phi$$

Optimal design of experiments

For fixed $\boldsymbol{x}$, the covariance of $\hat{\boldsymbol{a}}$ is

$$Cov(\hat{\boldsymbol{a}}) = (\Phi^T \boldsymbol{W} \Phi)^{-1} \Phi^T \boldsymbol{W} \Lambda \boldsymbol{W} \Phi (\Phi^T \boldsymbol{W} \Phi)^{-1}$$

For $\Lambda = \lambda \boldsymbol{W}^{-1}$, we obtain

$$Cov(\hat{\boldsymbol{a}}) = \lambda \boldsymbol{G}^{-1}$$

and the variance of the prediction $\hat{f}_m(x)$ at some point x is

$$\mathbb{V}(\hat{f}_m(x)) = \lambda \boldsymbol{\varphi}(x)^T \boldsymbol{G}^{-1} \boldsymbol{\varphi}(x)$$

In order to minimize the variance for any $x \in \mathcal{X}$, that is for any $\boldsymbol{\varphi}(x) \in \mathbb{R}^m$, we would like to minimize $\boldsymbol{G}^{-1}$ over $\boldsymbol{x} \in \mathcal{X}^n$ and $\boldsymbol{w} \in \mathbb{R}_+^n$, in the sense of the Loewner order, over the space S_m^+ of symmetric positive semi-definite matrices. However, a global optimum does not necessarily exist since Loewner order is only a partial order.

Optimal design of experiments

A common approach is to consider as a proxy the minimization of a decreasing convex function $h : S_m^+ \to \mathbb{R}$, i.e. such that

$$h(\boldsymbol{A}) \leq h(\boldsymbol{B}) \quad \text{for} \quad \boldsymbol{A} \succcurlyeq \boldsymbol{B},$$

and solve

$$\min_{\boldsymbol{x}, \boldsymbol{w}} h(\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{w}))$$

- E-optimal design corresponds $h(\boldsymbol{G}) = \lambda_{max}(\boldsymbol{G}^{-1}) = \lambda_{min}(\boldsymbol{G})^{-1}$
- A-optimal design corresponds to $h(\boldsymbol{G}) = Tr(\boldsymbol{G}^{-1})$
- D-optimal design corresponds to $h(\boldsymbol{G}) = det(\boldsymbol{G}^{-1}) = det(\boldsymbol{G})^{-1}$
- c-optimal design correspond to $h(\boldsymbol{G}) = \boldsymbol{c}^T \boldsymbol{G}^{-1} \boldsymbol{c}$ for some vector $\boldsymbol{c} \in \mathbb{R}^m$.

Least-squares approximation with i.i.d. sampling

Assume that the x_i are i.i.d. samples from a distribution $d\nu(x) = w(x)^{-1} d\mu(x)$ for some weight function w, and $w_i = w(x_i)$. We have

$$\boldsymbol{G} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{A}_i, \quad \boldsymbol{A}_i = w(x_i) \boldsymbol{\varphi}(x_i) \boldsymbol{\varphi}(x_i)^T,$$

where the $\boldsymbol{A}_i$ are i.i.d. rank-one matrices with expectation

$$\mathbb{E}(\boldsymbol{A}_i) = \mathbb{E}_{x \sim \nu}(w(x) \boldsymbol{\varphi}(x) \boldsymbol{\varphi}(x)^T) = \mathbb{E}_{x \sim \mu}(\boldsymbol{\varphi}(x) \boldsymbol{\varphi}(x)^T) = \boldsymbol{I}$$

and spectral norm

$$\|\boldsymbol{A}_i\| = w(x_i) \|\boldsymbol{\varphi}(x_i)\|_2^2 \leq K_{w,m},$$

with

$$K_{w,m} = \sup_{x \in \mathcal{X}} w(x) \|\boldsymbol{\varphi}(x)\|_2^2.$$

Least-squares approximation with i.i.d. sampling

Assume that the x_i are i.i.d. samples from a distribution $d\nu(x) = w(x)^{-1} d\mu(x)$ for some weight function w, and $w_i = w(x_i)$. We have

$$\boldsymbol{G} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{A}_i, \quad \boldsymbol{A}_i = w(x_i) \boldsymbol{\varphi}(x_i) \boldsymbol{\varphi}(x_i)^T,$$

where the $\boldsymbol{A}_i$ are i.i.d. rank-one matrices with expectation

$$\mathbb{E}(\boldsymbol{A}_i) = \mathbb{E}_{x \sim \nu}(w(x) \boldsymbol{\varphi}(x) \boldsymbol{\varphi}(x)^T) = \mathbb{E}_{x \sim \mu}(\boldsymbol{\varphi}(x) \boldsymbol{\varphi}(x)^T) = \boldsymbol{I}$$

and spectral norm

$$\|\boldsymbol{A}_i\| = w(x_i) \|\boldsymbol{\varphi}(x_i)\|_2^2 \leq K_{w,m},$$

with

$$K_{w,m} = \sup_{x \in \mathcal{X}} w(x) \|\boldsymbol{\varphi}(x)\|_2^2.$$

Based on matrix Chernoff concentration inequality, it can be shown that for any $0 < \delta < 1$,

$$\mathbb{P}(\lambda_{max}(\boldsymbol{G}) > 1 + \delta) \wedge \mathbb{P}(\lambda_{min}(\boldsymbol{G}) < 1 - \delta) \leq m \exp(-\frac{n\delta^2}{K_{w,m}})$$

and

$$\mathbb{P}(\|\boldsymbol{G} - \boldsymbol{I}\| > \delta) = \mathbb{P}(\lambda_{max}(\boldsymbol{G}) > 1 + \delta \text{ or } \lambda_{min}(\boldsymbol{G}) < 1 - \delta) \leq 2m \exp(-\frac{n\delta^2}{K_{w,m}})$$

Least-squares approximation with i.i.d. sampling

We obtain that

$$\mathbb{P}(\|\boldsymbol{G} - \boldsymbol{I}\| > \delta) \leq \eta$$

provided that

$$n \geq K_{w,m}\delta^{-2}\log(2m\eta^{-1}).$$

We note that

$$K_{w,m} = \sup_{x\in\mathcal{X}} w(x)\|\boldsymbol{\varphi}(x)\|_2^2 \geq \mathbb{E}_{x\sim\nu}(w(x)\|\boldsymbol{\varphi}(x)\|_2^2) = \mathbb{E}_{x\sim\mu}(\sum_{j=1}^{m}\varphi_j(x)^2)$$

so that

$$K_{w,m} \geq m$$

Classical least-squares approximation with i.i.d. sampling

For classical least-squares, $w = 1$ ($\nu = \mu$).

- For V_m piecewise constant functions on a uniform partition of $(0,1)$ and μ the uniform measure, $\boxed{K_{1,m} = m}$.
- For V_m trigonometric polynomials of degree $(m-1)/2$ on $(0, 2\pi)$ and μ the uniform measure, $\boxed{K_{1,m} = m}$.
- For polynomial spaces $V_m = \mathbb{P}_{m-1}$ and μ the uniform measure, $\boxed{K_{1,m} = m^2}$.
- For polynomial spaces $V_m = \mathbb{P}_{m-1}$ and μ the gaussian measure on $\mathbb{R}$, $\boxed{K_{1,m} = \infty}$.

Optimal weighted least squares with i.i.d. sampling

With i.i.d. sampling, an optimal sampling measure ν_m is given by $d\nu_m(x) = w_m(x)^{-1} d\mu(x)$ with density

$$w_m(x)^{-1} = \frac{1}{m} \sum_{j=1}^{m} \varphi_j(x)^2$$

that minimizes $K_{w,m}$ over all densities, and yields

$$K_{w_m,m} = m.$$

For polynomial approximation, $\sum_{j=1}^{m} \varphi_j(x)^2$ is the inverse of the Christoffel function.

Under the condition

$$n \geq m\delta^{-2} \log(2m\eta^{-1})$$

we have

$$\mathbb{P}(\|\boldsymbol{G} - \boldsymbol{I}\| > \delta) \leq \eta$$

Optimal weighted least squares with i.i.d. sampling

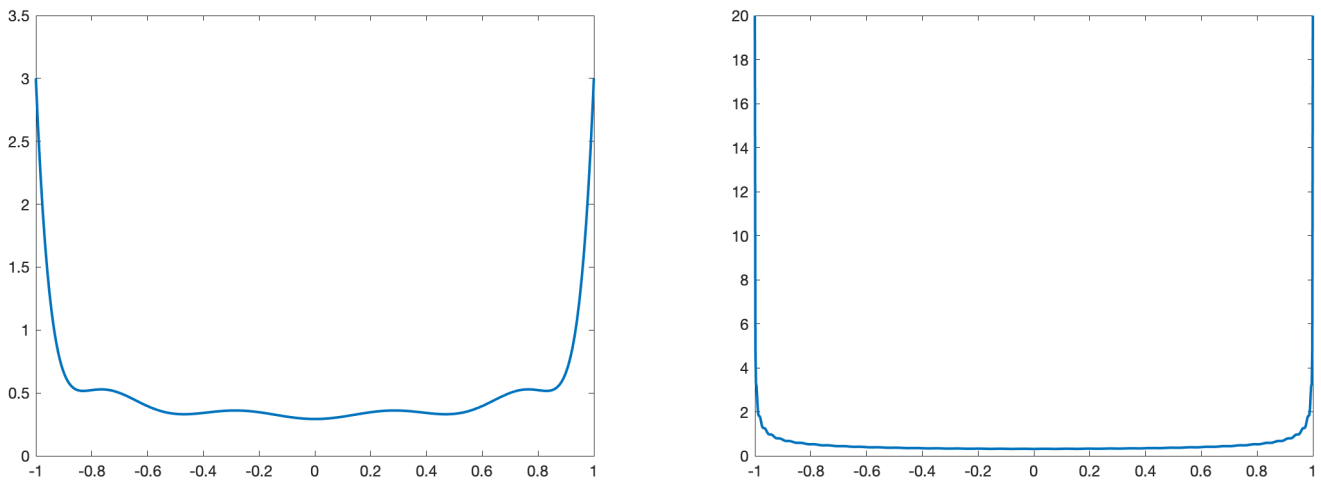
- For V_m piecewise constant functions on a uniform partition of $(0, 1)$ and μ the uniform measure, $w_m(x) = 1$.

Optimal weighted least squares with i.i.d. sampling

- For V_m piecewise constant functions on a uniform partition of $(0,1)$ and μ the uniform measure, $w_m(x) = 1$.
- For V_m trigonometric polynomials of degree $(m-1)/2$ on $(0, 2\pi)$ and μ the uniform measure, $w_m(x) = 1$.

Optimal weighted least squares with i.i.d. sampling

- For V_m piecewise constant functions on a uniform partition of $(0,1)$ and μ the uniform measure, $w_m(x) = 1$.
- For V_m trigonometric polynomials of degree $(m-1)/2$ on $(0, 2\pi)$ and μ the uniform measure, $w_m(x) = 1$.
- For polynomial spaces $V_m = \mathbb{P}_{m-1}$ and μ the uniform measure on $(-1,1)$

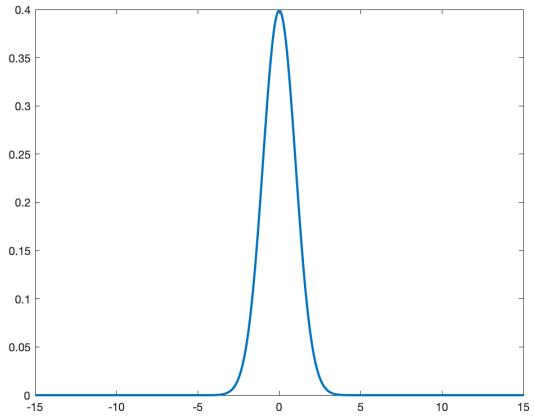


(a) $m = 6$ (b) $m = 40$

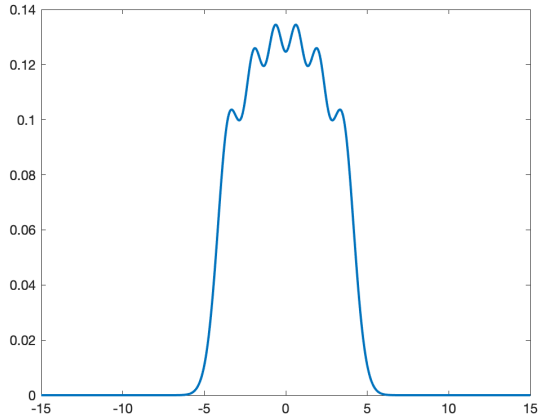
Figure: Polynomials and uniform measure: density of ν_m

Optimal weighted least squares with i.i.d. sampling

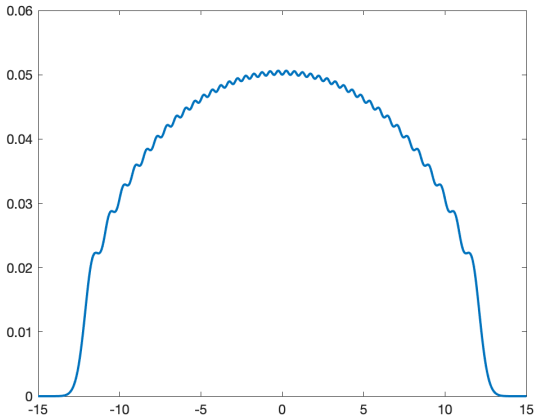
- For polynomial spaces $V_m = \mathbb{P}_{m-1}$ and μ the gaussian measure on $\mathbb{R}$



(a) $m = 0$



(b) $m = 6$



(c) $m = 40$

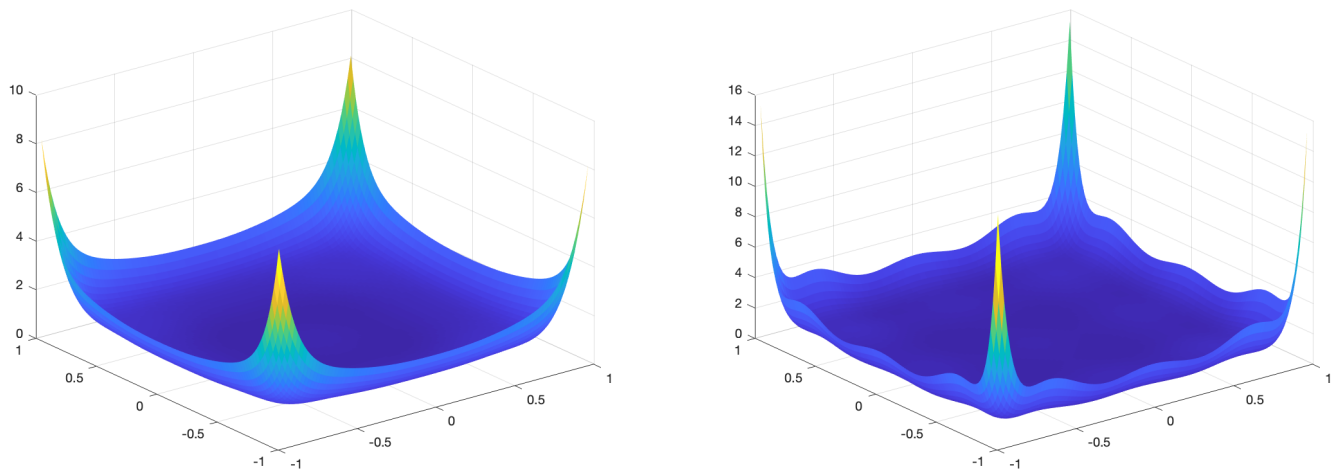
Figure: Polynomials and Gaussian measure: density of ν_m

Optimal weighted least squares with i.i.d. sampling

- For *d*-variate polynomials,

$$V_m = \mathbb{P}_\Lambda := span\{x^\alpha = x_1^{\alpha_1} \dots x_d^{\alpha_d} : \nu \in \Lambda \subset \mathbb{N}^d\}$$

$\Lambda = \Lambda_{1,p} := \{\alpha : \|\alpha\|_1 \leq p\}$ corresponds to polynomials with total degree $\leq p$.
$\Lambda = \Lambda_{\infty,p} := \{\alpha : \|\alpha\|_\infty \leq p\}$ corresponds to polynomials with partial degree $\leq p$.



(a) $\Lambda_{1,4}$ (b) $\Lambda_{\infty,4}$

Figure: Polynomials and uniform measure on $[-1,1]^2$:
density w_m for polynomials with total (left) or partial (right) degree less than 4.

Sampling from the optimal measure

We have to sample from the optimal measure

$$d\nu_m = w_m^{-1} d\mu, \quad w_m(x)^{-1} = \frac{1}{m} \sum_{j=1}^{m} \varphi_j(x)^2$$

Standard sampling technique can be used: inverse transform, rejection, Markov Chain Monte-Carlo...

However, for general spaces V_m, sampling may be a non trivial task.

Sampling from the optimal measure: mixture sampling

We observe that ν_m is a mixture of measures

$$d\nu^{(j)}(x) = \varphi_j(x)^2 d\mu(x)$$

with equal weights $1/m$. We can first sample j uniformly at random in $\{1, \ldots, m\}$ and then sample from $\nu^{(j)}$.

Recycling samples for adaptive approximation

In adaptive approximation, we construct approximations from a sequence of spaces $(V_m)_{m \geq 1}$.

To each space V_m is associated a specific optimal sampling measure $\nu_m = w_m^{-1} \mu$.

When functions evaluations are costly, we would like to exploit samples generated at previous iterations.

Recycling samples for adaptive approximation: hierarchical spaces

Consider the adaptive approximation in a sequence of nested spaces

$$V_1 \subset \ldots \subset V_m \subset V_{m+1} \subset \ldots$$

Let $(\varphi_j)_{j\geq 1}$ be such that $V_m = span\{\varphi_1, \ldots, \varphi_m\}$. Then

$$V_{m+1} = V_m \oplus span\{\varphi_{m+1}\}$$

and the optimal sampling measure ν_{m+1} associated to V_{m+1} is such that

$$d\nu_{m+1}(x) = \frac{1}{m+1}\sum_{j=1}^{m+1}\varphi_j(x)^2 d\mu(x) = \frac{m}{m+1}d\nu_m(x) + \frac{1}{m+1}\varphi_{m+1}^2 d\mu(x)$$

that corresponds to a mixture between ν_m and $\varphi_{m+1}^2\mu$, with respective weights $\frac{m}{m+1}$ and $\frac{1}{m+1}$.

Recycling samples for adaptive approximation: hierarchical spaces

Consider the adaptive approximation in a sequence of nested spaces

$$V_1 \subset \ldots \subset V_m \subset V_{m+1} \subset \ldots$$

Let $(\varphi_j)_{j\geq 1}$ be such that $V_m = span\{\varphi_1, \ldots, \varphi_m\}$. Then

$$V_{m+1} = V_m \oplus span\{\varphi_{m+1}\}$$

and the optimal sampling measure ν_{m+1} associated to V_{m+1} is such that

$$d\nu_{m+1}(x) = \frac{1}{m+1}\sum_{j=1}^{m+1}\varphi_j(x)^2 d\mu(x) = \frac{m}{m+1}d\nu_m(x) + \frac{1}{m+1}\varphi_{m+1}^2 d\mu(x)$$

that corresponds to a mixture between ν_m and $\varphi_{m+1}^2\mu$, with respective weights $\frac{m}{m+1}$ and $\frac{1}{m+1}$.

To sample the mixture, draw a Bernoulli variable $B(\frac{1}{m+1})$. If 1 is obtained, generate a new sample from $\varphi_{m+1}^2\mu$. If 0 is obtain, then either pick without replacement a sample from previously generated samples from ν_m, or generate a new sample from ν_m.

Different strategies can be found in [Arras et al 2019, Migliorati 2019].

Optimal weighted least-squares: error analysis

Let $f_m = P_{V_m} f$ be the orthogonal projection of f onto V_m w.r.t. the norm $\|\cdot\|$, that is the element of best approximation of f in V_m.
We have

$$\begin{aligned}\|f - \hat{f}_m\|^2 &\leq \|f - f_m\|^2 + \|f_m - \hat{f}_m\|^2 \\ &\leq \|f - f_m\|^2 + \lambda_{min}(\boldsymbol{G})^{-1}\|f_m - \hat{f}_m\|_n^2 \\ &\leq \|f - f_m\|^2 + \lambda_{min}(\boldsymbol{G})^{-1}\|f_m - f\|_n^2\end{aligned}$$

where we have used the fact that $\hat{f}_m$ is the orthogonal projection of f onto V_m w.r.t. the semi-norm $\|\cdot\|_n$.

Optimal weighted least-squares: error analysis

Let $f_m = P_{V_m} f$ be the orthogonal projection of f onto V_m w.r.t. the norm $\|\cdot\|$, that is the element of best approximation of f in V_m.
We have

$$\begin{aligned}\|f - \hat{f}_m\|^2 &\leq \|f - f_m\|^2 + \|f_m - \hat{f}_m\|^2 \\ &\leq \|f - f_m\|^2 + \lambda_{min}(\boldsymbol{G})^{-1}\|f_m - \hat{f}_m\|_n^2 \\ &\leq \|f - f_m\|^2 + \lambda_{min}(\boldsymbol{G})^{-1}\|f_m - f\|_n^2\end{aligned}$$

where we have used the fact that $\hat{f}_m$ is the orthogonal projection of f onto V_m w.r.t. the semi-norm $\|\cdot\|_n$.

If $\|\boldsymbol{G} - \boldsymbol{I}\| \leq \delta$, then $\lambda_{min}(\boldsymbol{G}) \geq 1 - \delta$ and

$$\|f - \hat{f}_m\|^2 \leq \|f - f_m\|^2 + (1 - \delta)^{-1}\|f - f_m\|_n^2$$

Optimal weighted least-squares: error analysis

Let $f_m = P_{V_m} f$ be the orthogonal projection of f onto V_m w.r.t. the norm $\|\cdot\|$, that is the element of best approximation of f in V_m.
We have

$$\begin{aligned}\|f - \hat{f}_m\|^2 &\leq \|f - f_m\|^2 + \|f_m - \hat{f}_m\|^2 \\ &\leq \|f - f_m\|^2 + \lambda_{min}(\boldsymbol{G})^{-1}\|f_m - \hat{f}_m\|_n^2 \\ &\leq \|f - f_m\|^2 + \lambda_{min}(\boldsymbol{G})^{-1}\|f_m - f\|_n^2\end{aligned}$$

where we have used the fact that $\hat{f}_m$ is the orthogonal projection of f onto V_m w.r.t. the semi-norm $\|\cdot\|_n$.

If $\|\boldsymbol{G} - \boldsymbol{I}\| \leq \delta$, then $\lambda_{min}(\boldsymbol{G}) \geq 1 - \delta$ and

$$\|f - \hat{f}_m\|^2 \leq \|f - f_m\|^2 + (1 - \delta)^{-1}\|f - f_m\|_n^2$$

In order to control of the approximation when $\|\boldsymbol{G} - \boldsymbol{I}\| > \delta$, different alternatives:

- assuming $\|f\|_\infty \leq \tau$, define a truncated estimator $\hat{f}_m^\tau = T_\tau \circ \hat{f}_m$ with $T_\tau(t) = sign(t)\min\{|t|, \tau\}$,
- define a conditional estimator $\hat{f}_m^C = f_m$ if $\|\boldsymbol{G} - \boldsymbol{I}\| \leq \delta$ or 0 if $\|\boldsymbol{G} - \boldsymbol{I}\| > \delta$,
- condition the samples to guarantee stability $\|\boldsymbol{G} - \boldsymbol{I}\|$.

Optimal weighted least-squares with conditioning

Assume that $\boldsymbol{x} = (x_1, \ldots, x_n)$ are drawn from $\nu^{\otimes n}$ conditioned to satisfy the event $S = \{\|\boldsymbol{G}(\boldsymbol{x}) - \boldsymbol{I}\| \leq \delta\}$. This can be obtained by sampling $\boldsymbol{x}$ from $\nu^{\otimes n}$ until S is satisfied (rejection).

Under the condition

$$n \geq m\delta^{-2} \log(2m\eta^{-1}) \tag{3}$$

we have

$$\mathbb{P}(S) \geq 1 - \eta$$

For $\eta < 1$, the random number N of samples from $\nu^{\otimes n}$ generated before acceptation follows a geometric distribution with parameter $\mathbb{P}(S)$, is almost surely finite, and with expectation $\mathbb{E}(N) = \mathbb{P}(S)^{-1} \leq (1 - \eta)^{-1}$.

Optimal weighted least-squares with conditioning

Assume that $\boldsymbol{x} = (x_1, \ldots, x_n)$ are drawn from $\nu^{\otimes n}$ conditioned to satisfy the event $S = \{\|\boldsymbol{G}(\boldsymbol{x}) - \boldsymbol{I}\| \leq \delta\}$. This can be obtained by sampling $\boldsymbol{x}$ from $\nu^{\otimes n}$ until S is satisfied (rejection).

Under the condition

$$n \geq m\delta^{-2} \log(2m\eta^{-1}) \tag{3}$$

we have

$$\mathbb{P}(S) \geq 1 - \eta$$

For $\eta < 1$, the random number N of samples from $\nu^{\otimes n}$ generated before acceptation follows a geometric distribution with parameter $\mathbb{P}(S)$, is almost surely finite, and with expectation $\mathbb{E}(N) = \mathbb{P}(S)^{-1} \leq (1 - \eta)^{-1}$.

The least-squares estimator satisfies

$$\begin{aligned}
\mathbb{E}(\|f - \hat{f}_m\|^2) &\leq \|f - f_m\|^2 + (1 - \delta)^{-1}\mathbb{E}(\|f - f_m\|_n^2) \\
&\leq \|f - f_m\|^2 + (1 - \delta)^{-1}(1 - \eta)^{-1}\mathbb{E}_{\boldsymbol{x} \sim \nu^{\otimes n}}(\|f - f_m\|_n^2) \\
&= (1 + (1 - \delta)^{-1}(1 - \eta)^{-1})\|f - f_m\|^2
\end{aligned}$$

Optimal weighted least-squares with conditioning

Therefore, we deduce a quasi-optimality in expectation

$$\mathbb{E}(\|f - \hat{f}_m\|^2)^{1/2} \leq C \inf_{v \in V_m} \|f - v\|,$$

with $C = (1 + (1-\delta)^{-1}(1-\eta)^{-1})^{1/2}$.

Optimal weighted least-squares with conditioning

Therefore, we deduce a quasi-optimality in expectation

$$\mathbb{E}(\|f - \hat{f}_m\|^2)^{1/2} \leq C \inf_{v \in V_m} \|f - v\|,$$

with $C = (1 + (1-\delta)^{-1}(1-\eta)^{-1})^{1/2}$.

For a compact set K of functions in L^2_μ, using the previous result with an optimal subspace V_m of dimension m such that

$$\inf_{v \in V_m} \|f - v\| = d_m(K)_{L^2_\mu},$$

we deduce that for $n \gtrsim cm\log(m)$, for some universal constant c, there exists a distribution over $\mathcal{X}^n$ and a linear recovery map A such that

$$\mathbb{E}(\|f - A(f(x_1), \ldots, f(x_n))\|^2)^{1/2} \leq C d_m(K)_{L^2_\mu}$$

which proves

$$\rho^{rand}_{cm\log(m)}(K)_{L^2_\mu} \leq C d_m(K)_{L^2_\mu}$$

Optimal weighted least-squares with conditioning and subsampling

By conditioning, we obtain $n \geq cm \log(m)$ samples that guarantee almost surely

$$\|\boldsymbol{G} - \boldsymbol{I}\| \leq \delta$$

However, the number of samples n may be large compared to m, and a fundamental question is whether the $\log(m)$ factor can be removed.

Optimal weighted least-squares with conditioning and subsampling

By conditioning, we obtain $n \geq cm\log(m)$ samples that guarantee almost surely

$$\|\boldsymbol{G} - \boldsymbol{I}\| \leq \delta$$

However, the number of samples n may be large compared to m, and a fundamental question is whether the $\log(m)$ factor can be removed.

In [Haberstich, Nouy and Perrin 2022], a subsampling approach is proposed, which consists in removing samples until the stability condition is violated.
More precisely, for $I \subset \{1, \dots, n\}$, we let $\boldsymbol{G}_I = \frac{1}{|I|}\sum_{i\in I} \boldsymbol{A}_i$. Starting from the set $I = \{1, \dots, n\}$, we successively remove from the current set I an index i such that

$$i \in \min_{j\in I} \|\boldsymbol{G}_{I\setminus\{j\}} - \boldsymbol{I}\|$$

If $\|\boldsymbol{G}_{I\setminus\{i\}} - \boldsymbol{I}\| > \delta$, we stop and return I. Otherwise, we continue removing samples.

Optimal weighted least-squares with conditioning and subsampling

By conditioning, we obtain $n \geq cm\log(m)$ samples that guarantee almost surely

$$\|\boldsymbol{G} - \boldsymbol{I}\| \leq \delta$$

However, the number of samples n may be large compared to m, and a fundamental question is whether the $\log(m)$ factor can be removed.

We observe in many applications that the algorithm returns a number of samples close to or even equal to m, without any theoretical guaranty.

Optimal weighted least-squares with conditioning and subsampling

In [Cohen and Dolbeault 2021], it is proposed a subsampling strategy, based on successive random partitioning of the set of samples, which yields a number of samples in $O(m)$ while preserving stability.[1]
Note that

$$\boldsymbol{G} = \sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{a}_i^T \quad \text{with} \quad \boldsymbol{a}_i = \sqrt{\frac{w(x_i)}{n}} \boldsymbol{\varphi}(x_i) \in \mathbb{R}^m.$$

We have

$$(1-\delta)\boldsymbol{I} \preccurlyeq \sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{a}_i^T \preccurlyeq (1+\delta)\boldsymbol{I} \quad \text{and} \quad \|\boldsymbol{a}_i\|_2^2 = m/n.$$

[1] It relies on results from [Markus, Spielman and Srivastava 2015][Nitzan, Olevskii and Olevskii 2016] that provide a solution to the Kadinson-Singer problem.

Optimal weighted least-squares with conditioning and subsampling

In [Cohen and Dolbeault 2021], it is proposed a subsampling strategy, based on successive random partitioning of the set of samples, which yields a number of samples in $O(m)$ while preserving stability.[1]
Note that

$$\boldsymbol{G} = \sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{a}_i^T \quad \text{with} \quad \boldsymbol{a}_i = \sqrt{\frac{w(x_i)}{n}} \boldsymbol{\varphi}(x_i) \in \mathbb{R}^m.$$

We have

$$(1-\delta)\boldsymbol{I} \preccurlyeq \sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{a}_i^T \preccurlyeq (1+\delta)\boldsymbol{I} \quad \text{and} \quad \|\boldsymbol{a}_i\|_2^2 = m/n.$$

A procedure is introduced which provides a partition of $\{1, \ldots, n\}$ into sets $J_1, \ldots, J_{2^L}$ with cardinal $|J_k| \leq cm$, and such that for all $1 \leq k \leq 2^L$

$$c_0 \boldsymbol{I} \preccurlyeq \frac{n}{m} \sum_{i \in J_k} \boldsymbol{a}_i \boldsymbol{a}_i^T \preccurlyeq C_0 \boldsymbol{I}$$

with universal constants c_0 and C_0. Then pick k at random in $\{1, \ldots, 2^L\}$ with probability $p_k = |J_k|/m$.

[1] It relies on results from [Markus, Spielman and Srivastava 2015][Nitzan, Olevskii and Olevskii 2016] that provide a solution to the Kadinson-Singer problem.

Optimal weighted least-squares with conditioning and subsampling

This proves that

$$\rho_{cm}^{rand}(K)_{L^2_\mu} \leq C d_m(K)_{L^2_\mu}$$

for some universal constants c and C.

However, the subsampling strategy is not computationally feasible.

Other subsampling strategy have been proposed in [Bartel et al 2022], with theoretical guarantees and feasible implementations.

Optimal weighted least-squares with conditioning and subsampling

Note that the samples $x_1, \ldots, x_n$ obtained by conditioning (and possibly subsampling) are no more independent and follows a distribution which is not explicit.

In adaptive setting, we can no more recycle samples using mixture sampling.

An alternative recycling method has been proposed in [Haberstich 2020].

Control in probability

We would like to obtain quasi-optimality guarantees with high probability, or even almost surely, for the approximation of functions from a space X continuously embedded in L^2_μ, that is such that $\|f\| \leq C_X \|f\|_X$ for all $f \in X$.

For that, the sampling should depend on both X and V_m.

Control in probability

We would like to obtain quasi-optimality guarantees with high probability, or even almost surely, for the approximation of functions from a space X continuously embedded in L^2_μ, that is such that $\|f\| \leq C_X \|f\|_X$ for all $f \in X$.

For that, the sampling should depend on both X and V_m.

We can consider a mixture between the optimal distribution $w_m^{-1} d\mu$ and a distribution $h d\mu$, with density

$$w(x)^{-1} = \frac{1}{2} w_m(x)^{-1} + \frac{1}{2} h(x),$$

where h is a related to X.

Control in probability

We would like to obtain quasi-optimality guarantees with high probability, or even almost surely, for the approximation of functions from a space X continuously embedded in L^2_μ, that is such that $\|f\| \leq C_X \|f\|_X$ for all $f \in X$.

For that, the sampling should depend on both X and V_m.

We can consider a mixture between the optimal distribution $w_m^{-1} d\mu$ and a distribution $h d\mu$, with density

$$w(x)^{-1} = \frac{1}{2} w_m(x)^{-1} + \frac{1}{2} h(x),$$

where h is a related to X.

The empirical Gram matrix $\boldsymbol{G}$ remains an unbiased estimator of $\boldsymbol{I}$ and

$$K_{w,m} = \sup_{x \in \mathcal{X}} w(x) \|\varphi(x)\|_2^2 \leq 2K_{w_m,m} = 2m$$

Therefore, only a factor 2 is lost in the number of samples required to ensure $\|\boldsymbol{G} - \boldsymbol{I}\| \leq \delta$ with nonzero probability. By conditioning we obtain almost surely the error bound

$$\|f - \hat{f}_m\| \leq \|f - g\| + (1 - \delta)^{-1/2} \|f - g\|_n \quad \forall g \in V_m.$$

Control in probability

We would like to obtain quasi-optimality guarantees with high probability, or even almost surely, for the approximation of functions from a space X continuously embedded in L^2_μ, that is such that $\|f\| \leq C_X \|f\|_X$ for all $f \in X$.

For that, the sampling should depend on both X and V_m.

We can consider a mixture between the optimal distribution $w_m^{-1} d\mu$ and a distribution $h d\mu$, with density

$$w(x)^{-1} = \frac{1}{2} w_m(x)^{-1} + \frac{1}{2} h(x),$$

where h is a related to X.

The empirical Gram matrix $\boldsymbol{G}$ remains an unbiased estimator of $\boldsymbol{I}$ and

$$K_{w,m} = \sup_{x \in \mathcal{X}} w(x) \|\varphi(x)\|_2^2 \leq 2 K_{w_m,m} = 2m$$

Therefore, only a factor 2 is lost in the number of samples required to ensure $\|\boldsymbol{G} - \boldsymbol{I}\| \leq \delta$ with nonzero probability. By conditioning we obtain almost surely the error bound

$$\|f - \hat{f}_m\| \leq \|f - g\| + (1-\delta)^{-1/2} \|f - g\|_n \quad \forall g \in V_m.$$

If the function h is chosen such that for all $f \in X$, $\|f\|_n \leq C \|f\|_X$, we obtain

$$\|f - \hat{f}_m\| \leq (C_X + (1-\delta)^{-1/2} C) \inf_{g \in V_m} \|f - g\|_X$$

Control in probability

For $X = L^\infty_\mu(\mathcal{X})$ equipped with its natural norm $\|\cdot\|_\infty$, we can take

$$h(x) = 1$$

so that $w(x)^{-1} \geq 1/2$. For all $f \in X$, we then have $\|f\| \leq \|f\|_\infty$ and

$$\|f\|_n^2 = \frac{1}{n}\sum_{i=1}^n w(x_i) f(x_i)^2 \leq \frac{2}{n}\sum_{i=1}^n f(x_i)^2 \leq 2\|f\|_\infty^2$$

This yields

$$\|f - \hat{f}_m\| \leq (1 + (1-\delta)^{-1/2}\sqrt{2}) \inf_{g \in V_m} \|f - g\|_\infty$$

Control in probability

Consider for X a RKHS with a kernel k in $L^2_{\mu\otimes\mu}(\mathcal{X}\times\mathcal{X})$ that admits a decomposition

$$k(x,y)=\sum_{i\geq 1}\lambda_i\psi_i(x)\psi_i(y)$$

where the ψ_i form an orthonormal basis of $L^2_\mu(\mathcal{X})$ and where $(\lambda_i)_{i\geq 1}$ is a decreasing sequence of strictly positive numbers such that

$$\sum_{i\geq 1}\lambda_i^2=\|k\|^2_{L^2}<\infty.$$

The (ψ_i,λ_i) are the eigenpairs of the Hilbert-Schmidt integral operator T_k with kernel k.

Control in probability

Consider for X a RKHS with a kernel k in $L^2_{\mu\otimes\mu}(\mathcal{X}\times\mathcal{X})$ that admits a decomposition

$$k(x,y)=\sum_{i\geq 1}\lambda_i\psi_i(x)\psi_i(y)$$

where the ψ_i form an orthonormal basis of $L^2_\mu(\mathcal{X})$ and where $(\lambda_i)_{i\geq 1}$ is a decreasing sequence of strictly positive numbers such that

$$\sum_{i\geq 1}\lambda_i^2=\|k\|^2_{L^2}<\infty.$$

The (ψ_i,λ_i) are the eigenpairs of the Hilbert-Schmidt integral operator T_k with kernel k.

The norm on X is given by

$$\|f\|_X^2=\sum_{i\geq 1}(f,\psi_i)^2_{L^2_\mu}/\lambda_i,$$

and

$$\|f\|^2=\sum_{i\geq 1}(f,\psi_i)^2_{L^2_\mu}=\sum_{i\geq 1}\lambda_i(f,\psi_i)^2_{L^2_\mu}/\lambda_i\leq\lambda_1\|f\|_X^2.$$

Therefore, X is continuously embedded in L^2_μ with embedding constant $C_X=\lambda_1^{1/2}$.

Control in probability

We further assume (up to a rescaling) that

$$\sum_{i \geq 1} \lambda_i = \int k(x,x) d\mu(x) = 1 < \infty$$

that is T_k is nuclear (trace class) with unit nuclear norm.

Therefore, $k(x,x)$ defines a density and we can take

$$h(x) = k(x,x).$$

Control in probability

We further assume (up to a rescaling) that

$$\sum_{i \geq 1} \lambda_i = \int k(x,x) d\mu(x) = 1 < \infty$$

that is T_k is nuclear (trace class) with unit nuclear norm.

Therefore, $k(x,x)$ defines a density and we can take

$$h(x) = k(x,x).$$

We have $w(x)^{-1} \geq k(x,x)/2$, so that

$$\|f\|_n^2 \leq \frac{2}{n} \sum_{i=1}^n k(x_i,x_i)^{-1} f(x_i)^2 = \frac{2}{n} \sum_{i=1}^n k(x_i,x_i)^{-1} (k(x_i,\cdot), f)_X^2 \leq 2\|f\|_X^2$$

We finally deduce

$$\|f - \hat{f}_m\| \leq (\lambda_1 + (1-\delta)^{-1/2}\sqrt{2}) \inf_{g \in V_m} \|f - g\|_X$$

Sampling numbers

Using subsampling techniques from [Cohen and Dolbeault 2021], we then prove that for $X = L^\infty$ or X a RKHS associated with a trace class operator, there exists a set of $n \leq cm$ points and a linear algorithm such that for all $f \in X$, the produced approximation $\hat{f}_m = A(f(x_1), \ldots, f(x_n))$ is such that

$$\|f - \hat{f}_m\| \leq CE(f; V_m)_X$$

Sampling numbers

Using subsampling techniques from [Cohen and Dolbeault 2021], we then prove that for $X = L^\infty$ or X a RKHS associated with a trace class operator, there exists a set of $n \leq cm$ points and a linear algorithm such that for all $f \in X$, the produced approximation $\hat{f}_m = A(f(x_1), \ldots, f(x_n))$ is such that

$$\|f - \hat{f}_m\| \leq CE(f; V_m)_X$$

Consider a compact set $K \subset X$ and an optimal approximating subspace V_m in the sense that $\sup_{f \in K} E(f; V_m)_X = d_m(K)_X$. We then have proven that

$$\rho_{cm}(K)_{L^2} \leq Cd_m(K)_X$$

Sampling numbers

Using subsampling techniques from [Cohen and Dolbeault 2021], we then prove that for $X = L^\infty$ or X a RKHS associated with a trace class operator, there exists a set of $n \leq cm$ points and a linear algorithm such that for all $f \in X$, the produced approximation $\hat{f}_m = A(f(x_1), \dots, f(x_n))$ is such that

$$\|f - \hat{f}_m\| \leq CE(f; V_m)_X$$

Consider a compact set $K \subset X$ and an optimal approximating subspace V_m in the sense that $\sup_{f \in K} E(f; V_m)_X = d_m(K)_X$. We then have proven that

$$\rho_{cm}(K)_{L^2} \leq C d_m(K)_X$$

For K the unit ball of a RKHS (with the trace class assumption), a refined analysis (see [1]) yields

$$\rho_{cm}(K)_{L^2} \leq \sqrt{\frac{1}{m} \sum_{k \geq m} d_k(K)_{L^2}^2}$$

for some universal constant c, which is known as a sharp bound.

Sampling numbers

Using subsampling techniques from [Cohen and Dolbeault 2021], we then prove that for $X = L^\infty$ or X a RKHS associated with a trace class operator, there exists a set of $n \leq cm$ points and a linear algorithm such that for all $f \in X$, the produced approximation $\hat{f}_m = A(f(x_1), \ldots, f(x_n))$ is such that

$$\|f - \hat{f}_m\| \leq CE(f; V_m)_X$$

Consider a compact set $K \subset X$ and an optimal approximating subspace V_m in the sense that $\sup_{f \in K} E(f; V_m)_X = d_m(K)_X$. We then have proven that

$$\rho_{cm}(K)_{L^2} \leq C d_m(K)_X$$

For K the unit ball of a RKHS (with the trace class assumption), a refined analysis (see [1]) yields

$$\rho_{cm}(K)_{L^2} \leq \sqrt{\frac{1}{m} \sum_{k \geq m} d_k(K)_{L^2}^2}$$

for some universal constant c, which is known as a sharp bound. For a larger class of spaces including the space of bounded functions equipped with the supremum norm, they show

$$\rho_{cm}(K)_{L^2} \leq \left(\frac{1}{m} \sum_{k \geq m} d_k(K)_{L^2}^p \right)^{1/p} \qquad \text{for any } 0 < p < 2$$

Outline

1. Manifold approximation
2. Linear approximation from point evaluations
3. Tensor networks approximation with point evaluations

Algorithms for approximation with tensor networks

For the approximation of tensors (or functions) using tensor networks, different contexts depending on the available information:

- all entries of the tensor,
- equations satisfied by the tensor,
- some entries, either arbitrary or structured,
- more general functionals of the tensor.

Available packages

- tensap. A Python package for the approximation of functions and tensors. (link to GitHub page).
- ApproximationToolbox. An object-oriented MATLAB toolbox for the approximation of functions and tensors. (link to GitHub page).

Learning from structured evaluations

For the approximation of a multivariate function with tree tensor networks using point evaluations, different strategies have been proposed, either based on cross approximation [Oseledets'10, Ballani'13] or principal component analysis [Nouy'19, Haberstich'21].

These methods rely on structured evaluations

$$u(x_\alpha^i, x_{\alpha^c}^j)$$

where x_α^i are samples of the variables x_α, and $x_{\alpha^c}^j$ samples of the variables x_{α^c}.

Learning from principal component analysis

Consider a multivariate function $u \in L^2_\mu(\mathcal{X})$ where $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ is equipped with a probability measure $\mu = \mu_1 \otimes \ldots \otimes \mu_d$. Let $X = (X_1, \ldots, X_d)$ be a random vector with distribution μ, such that the L^2_μ-norm is given by

$$\|u\|^2 = \int u(x)^2 d\mu(x) = \mathbb{E}(u(X)^2).$$

For each a subset of variables α and its complementary subset $\alpha^c = D \setminus \alpha$, u is identified with a bivariate function defined on $\mathcal{X}_\alpha \times \mathcal{X}_{\alpha^c}$ which admits a singular value decomposition

$$u(x_\alpha, x_{\alpha^c}) = \sum_{k=1}^{\text{rank}_\alpha(u)} \sigma_k^\alpha v_k^\alpha(x_\alpha) v_k^{\alpha^c}(x_{\alpha^c})$$

Learning from principal component analysis

The subspace of α-principal components

$$U_\alpha = span\{v_1^\alpha, \ldots, v_{r_\alpha}^\alpha\}$$

is such that

$$u_{r_\alpha}(\cdot, x_{\alpha^c}) = \mathcal{P}_{U_\alpha} u(\cdot, x_{\alpha^c})$$

It is solution of

$$\min_{\dim(U_\alpha)=r_\alpha} \|u - \mathcal{P}_{U_\alpha} u\|^2$$

that is for $\|\cdot\|$ the $L^2_\mu(\mathcal{X})$-norm,

$$\min_{\dim(U_\alpha)=r_\alpha} \mathbb{E}\left(\|u(\cdot, X_{\alpha^c}) - \mathcal{P}_{U_\alpha} u(\cdot, X_{\alpha^c})\|^2_{L^2_{\mu_\alpha}(\mathcal{X}_\alpha)}\right)$$

where u is seen as a function-valued random variable

$$u(\cdot, X_{\alpha^c}) \in L^2_{\mu_\alpha}(\mathcal{X}_\alpha).$$

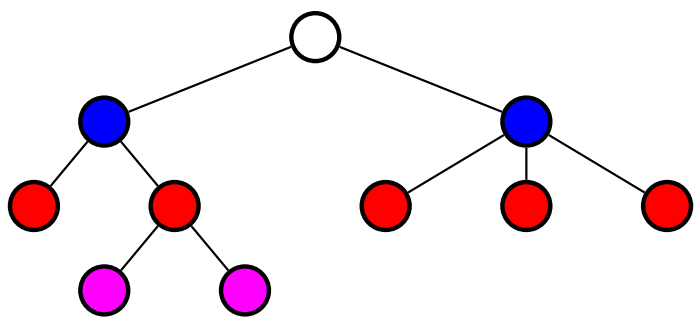
U_α is the optimal m-dimensional space for the approximation of the manifold $\{u(\cdot, x_{\alpha^c}) : x_{\alpha^c} \in \mathcal{X}_{\alpha^c}\}$ in mean-squared error.

Truncation scheme for tree-based tensor formats

For tree tensor networks

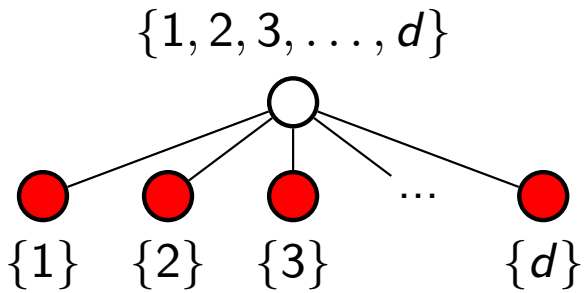
$$\mathcal{T}_r^T(V) = \{v \in V : \mathrm{rank}_\alpha(v) \leq r_\alpha, \alpha \in T\},$$

where T is a dimension partition tree over $D = \{1, \ldots, d\}$, different variants of higher order singular value decomposition (also called hierarchical singular value decomposition) can be defined from singular value decompositions of bivariate functions.



Higher-order principal component analysis for Tucker format

Tucker format corresponds a trivial tree with $d+1$ nodes (the root and the d leaves).



For each leaf $\nu \in \{1, \ldots, d\}$, we determine a $\{\nu\}$-principal subspace $U^{\nu}_{r_\nu}$ of dimension r_ν (a space of functions of the variable x_ν).

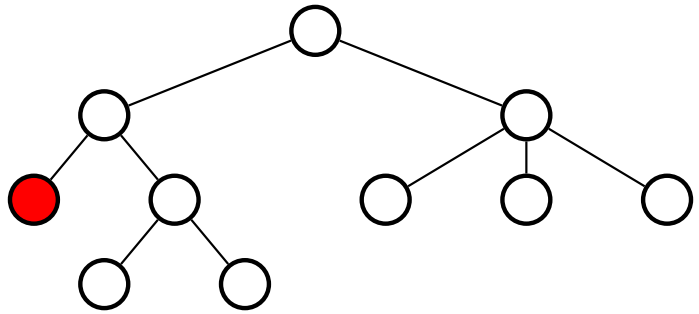
Then, we obtain an approximation in Tucker format (with ranks $r_1, \ldots, r_d$) by a projection of the function u onto the linear tensor product space

$$U_1 \otimes \ldots \otimes U_d$$

.

Leaves to root strategy for general tree tensor networks

For each leaf node α, let $U_{r_\alpha}^{\alpha}$ be the r_α-dimensional α-principal subspace of u.

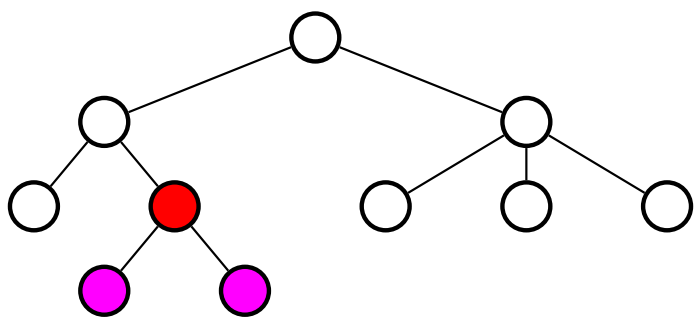


For each interior node $\alpha \in T \setminus \{D\}$ with children $S(\alpha)$, define a tensor space

$$V_\alpha = \bigotimes_{\beta \in S(\alpha)} U_{r_\beta}^{\beta}$$

and let $U_{r_\alpha}^{\alpha} \subset V_\alpha$ be the r_α-dimensional α-principal subspace of the function u_α defined by

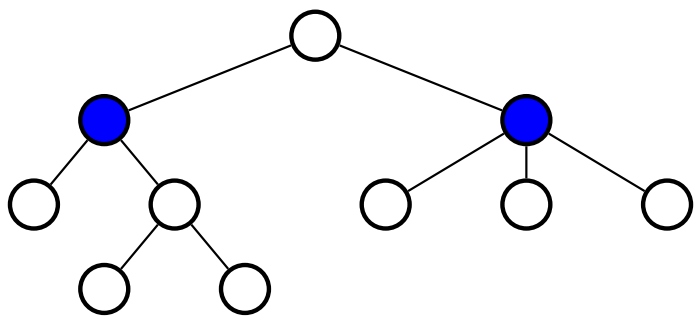
$$u_\alpha(\cdot, x_{\alpha^c}) = \mathcal{P}_{V_\alpha} u(\cdot, x_{\alpha^c})$$



Leaves to root strategy for general tree tensor networks

Finally define an approximation u_r as a projection of u onto the tensor space $V_D = \bigotimes_{\alpha \in S(D)} U_\alpha$.

We can prove that the resulting approximation u_r is a tree tensor network with ranks r_α, $\alpha \in T$.



Leaves to root truncation scheme for tree-based tensor formats

Provided we use orthogonal projections, the obtained approximation u_r is such that

$$\|u - u_r\|^2 \leq \sum_{\alpha \in T \setminus D} \min_{\mathrm{rank}_\alpha(v) \leq r_\alpha} \|u - v\|^2 = \sum_{\alpha \in T \setminus D} \sum_{k_\alpha > r_\alpha} (\sigma^\alpha_{k_\alpha})^2,$$

from which we deduce that u_r is a quasi-optimal approximation of u in $\mathcal{T}_r^T$ such that

$$\|u - u_r\| \leq C(T) \min_{v \in \mathcal{T}_r^T} \|u - v\|,$$

where $C(T) = \sqrt{\#T - 1}$ is the square root of the number of projections applied to the tensor. The number of nodes of a dimension partition tree T being bounded by $2d - 1$,

$$C(T) \leq \sqrt{2d - 2}.$$

Leaves to root truncation scheme for tree-based tensor formats

Provided we use orthogonal projections, the obtained approximation u_r is such that

$$\|u - u_r\|^2 \leq \sum_{\alpha \in T \setminus D} \min_{\text{rank}_\alpha(v) \leq r_\alpha} \|u - v\|^2 = \sum_{\alpha \in T \setminus D} \sum_{k_\alpha > r_\alpha} (\sigma^\alpha_{k_\alpha})^2,$$

from which we deduce that u_r is a quasi-optimal approximation of u in $\mathcal{T}_r^T$ such that

$$\|u - u_r\| \leq C(T) \min_{v \in \mathcal{T}_r^T} \|u - v\|,$$

where $C(T) = \sqrt{\#T - 1}$ is the square root of the number of projections applied to the tensor. The number of nodes of a dimension partition tree T being bounded by $2d - 1$,

$$C(T) \leq \sqrt{2d - 2}.$$

Also, if we select the ranks $(r_\alpha)_{\alpha \in T \setminus D}$ such that for all α

$$\sum_{k_\alpha > r_\alpha} (\sigma^\alpha_{k_\alpha})^2 \leq \frac{\epsilon^2}{C(T)^2} \sum_{k_\alpha \geq 1} (\sigma^\alpha_{k_\alpha})^2 = \frac{\epsilon^2}{C(T)^2} \|u\|^2,$$

we finally obtain an approximation u_r with relative precision ϵ,

$$\|u - u_r\| \leq \epsilon \|u\|.$$

Leaves to root truncation scheme for tree-based tensor formats

Given a finite dimensional tensor space $V = V_1 \otimes \ldots \otimes V_d$, an approximation in the tensor format $\mathcal{T}_r^T(V)$ can be obtained by modifying the procedure for the leaves.

For each leaf node α, $U_{r_\alpha}^\alpha$ is defined as a α-principal subspace of $u_\alpha = \mathcal{P}_{V_\alpha} u$.

Theorem (Fixed rank)

For a given T-rank, we obtain an approximation $u_r \in \mathcal{T}_r^T(V)$ such that

$$\|u_r - u\|^2 \leq C(T)^2 \min_{v \in \mathcal{T}_r^T} \|v - u\|^2 + \sum_{\text{leaves } \alpha} \|u - \mathcal{P}_{V_\alpha} u\|^2$$

Theorem (Fixed precision)

For a desired precision ϵ, if the α-ranks are determined such that

$$\|\mathcal{P}_{U_{r_\alpha}^\alpha} u_\alpha - u_\alpha\| \leq \frac{\epsilon}{C(T)} \|u_\alpha\|,$$

we obtain an approximation u_r such that

$$\|u_r - u\|^2 \leq \epsilon^2 \|u\|^2 + \sum_{\text{leaves } \alpha} \|u - \mathcal{P}_{V_\alpha} u\|^2.$$

Learning algorithm based on principal component analysis

For a feasible algorithm using samples:

- Replacement of orthogonal projections by sampled-based projections, based on interpolation [Nouy 2019] or optimal least-squares projections [Haberstich 2021].
- Statistical estimation of principal subspaces U_α by empirical PCA, using samples $u(\cdot, x^j_{\alpha^c})$

Learning algorithm based on principal component analysis

For a feasible algorithm using samples:

- Replacement of orthogonal projections by sampled-based projections, based on interpolation [Nouy 2019] or optimal least-squares projections [Haberstich 2021].
- Statistical estimation of principal subspaces U_α by empirical PCA, using samples $u(\cdot, x^j_{\alpha^c})$

The estimation of principal subspaces requires the evaluation of u on a structured set of points

$$\{(x^i_\alpha, x^j_{\alpha^c}) : 1 \leq i \leq M_\alpha, 1 \leq j \leq N_\alpha\}$$

where N_α is the number of samples $x^j_{\alpha^c}$ used for the estimation of U_α by empirical PCA, and M_α is the number of points x^i_α used for the projections onto the space V_α.

The sampling strategy is adaptive to the function.

Learning algorithm based on principal component analysis

For a feasible algorithm using samples:

- Replacement of orthogonal projections by sampled-based projections, based on interpolation [Nouy 2019] or optimal least-squares projections [Haberstich 2021].
- Statistical estimation of principal subspaces U_α by empirical PCA, using samples $u(\cdot, x^j_{\alpha^c})$

The estimation of principal subspaces requires the evaluation of u on a structured set of points

$$\{(x^i_\alpha, x^j_{\alpha^c}) : 1 \leq i \leq M_\alpha, 1 \leq j \leq N_\alpha\}$$

where N_α is the number of samples $x^j_{\alpha^c}$ used for the estimation of U_α by empirical PCA, and M_α is the number of points x^i_α used for the projections onto the space V_α.

The sampling strategy is adaptive to the function.

Some guarantees can be obtained under additional assumptions on the function to approximate [Haberstich 2021].

But yet not guaranty of quasi-optimality in a general setting.

Concluding remarks

Development of near optimal learning algorithms.

- Theory well established for least-squares approximation in linear spaces
- Mainly an open problem for linear approximation in other spaces than L^2
- Only partial results on optimal sampling for least-squares approximation with tensor networks, and mainly open problem for neural networks.
- Optimal sampling for manifold approximation ? Some results for linear manifold approximation (PCA, Reduced basis), but mainly an open problem for general nonlinear approximation of manifolds.

References I

Sampling and linear approximation

A. Cohen and G. Migliorati.

Optimal weighted least-squares methods.

SMAI Journal of Computational Mathematics, 3:181–203, 2017.

A. Cohen and M. Dolbeault.

Optimal pointwise sampling for l^2 approximation, 2021.

M. Dolbeault, D. Krieg, and M. Ullrich.

A sharp upper bound for sampling numbers in L_2.

arXiv e-prints, arXiv:2204.12621, Apr. 2022.

B. Arras, M. Bachmayr, and A. Cohen.

Sequential sampling for optimal weighted least squares approximations in hierarchical spaces.

SIAM Journal on Mathematics of Data Science, 1(1):189–207, 2019.

C. Haberstich, A. Nouy, and G. Perrin.

Boosted optimal weighted least-squares.

Mathematics of Computation, 91(335):1281–1315, 2022.

G. Migliorati.

Adaptive approximation by optimal weighted least-squares methods.

SIAM Journal on Numerical Analysis, 57(5):2217–2245, 2019.

References II

C. Haberstich.

Adaptive approximation of high-dimensional functions with tree tensor networks for Uncertainty Quantification.

Theses, École centrale de Nantes, Dec. 2020.

A. W. Marcus, D. A. Spielman, and N. Srivastava.

Interlacing families ii: Mixed characteristic polynomials and the kadison—singer problem.

Annals of Mathematics, pages 327–350, 2015.

S. Nitzan, A. Olevskii, and A. Olevskii.

Exponential frames on unbounded sets.

Proceedings of the American Mathematical Society, 144(1):109–118, 2016.

F. Bartel, M. Schäfer, and T. Ullrich.

Constructive subsampling of finite frames with applications in optimal function recovery.

arXiv preprint arXiv:2202.12625, 2022.

V. Temlyakov.

On optimal recovery in L2.

Journal of Complexity, 65:101545, 2021.

References III

N. Nagel, M. Schäfer, and T. Ullrich.

A new upper bound for sampling numbers.

Foundations of Computational Mathematics, pages 1–24, 2021.

Learning with tensor networks

B. Michel and A. Nouy.

Learning with tree tensor networks: complexity estimates and model selection.

arXiv e-prints, page arXiv:2007.01165, July 2020.

E. M. Stoudenmire and D. J. Schwab.

Supervised learning with quantum-inspired tensor networks, 2017.

E. Grelier, A. Nouy, M. Chevreuil.

Learning with tree-based tensor formats.

Arxiv eprints, Nov. 2018.

E. Grelier, A. Nouy, and R. Lebrun.

Learning high-dimensional probability distributions using tree tensor networks.

arXiv preprint arXiv:1912.07913, 2019.

A. Nouy.

Higher-order principal component analysis for the approximation of tensors in tree-based low-rank formats.

Numerische Mathematik, 141(3):743–789, Mar 2019.

References IV

C. Haberstich, A. Nouy, and G. Perrin.

Active learning of tree tensor networks using optimal least-squares.
arXiv preprint arXiv:2104.13436, 2021.

I. Oseledets and E. Tyrtyshnikov.

TT-cross approximation for multidimensional arrays.
Linear Algebra And Its Applications, 432(1):70–88, JAN 1 2010.

L. Grasedyck and S. Krämer.

Stable als approximation in the tt-format for rank-adaptive tensor completion.
Numerische Mathematik, 143(4):855–904, 2019.

Software

Nouy Anthony, Grelier Erwan and Giraldi Loic. (2020, February 7). ApproximationToolbox. Zenodo. http://doi.org/10.5281/zenodo.3653970

Anthony Nouy, & Erwan Grelier. (2020, June 15). anthony-nouy/tensap. Zenodo. http://doi.org/10.5281/zenodo.3894378

Manifold approximation

Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. H. Pau.

A general multipurpose interpolation procedure: the magic points.
Communications On Pure and Applied Analysis, 8(1):383–404, 2009.

References V

R. DeVore, G. Petrova, and P. Wojtaszczyk.

Greedy algorithms for reduced bases in banach spaces.

Constructive Approximation, 37(3):455–466, 2013.

O. Balabanov and A. Nouy.

Randomized linear algebra for model reduction. part i: Galerkin methods and error estimation.

Advances in Computational Mathematics, 45(5-6):2969–3019, 2019.

O. Balabanov and A. Nouy.

Randomized linear algebra for model reduction—part ii: minimal residual methods and dictionary-based approximation.

Advances in Computational Mathematics, 47(2):1–54, 2021.

O. Balabanov and A. Nouy.

Preconditioners for model order reduction by interpolation and random sketching of operators.

arXiv preprint arXiv:2104.12177, 2021.

A. Cohen, W. Dahmen, R. DeVore, and J. Nichols.

Reduced basis greedy selection using random training sets.

ESAIM: Mathematical Modelling and Numerical Analysis, 54(5):1509–1524, 2020.

References VI

M. Billaud-Friess, A. Macherey, A. Nouy, and C. Prieur.

A probabilistic reduced basis method for parameter-dependent problems.

In preparation, 2022.

A. Cohen, W. Dahmen, R. DeVore, and J. Nichols.

Reduced basis greedy selection using random training sets.

ESAIM: Mathematical Modelling and Numerical Analysis, 54(5):1509–1524, 2020.

J. L. Eftang, A. T. Patera, and E. M. Rønquist.

An "hp" certified reduced basis method for parametrized elliptic partial differential equations.

SIAM Journal on Scientific Computing, 32(6):3170–3200, 2010.

S. Kaulmann and B. Haasdonk.

Online greedy reduced basis construction using dictionaries.

In *VI International Conference on Adaptive Modeling and Simulation (ADMOS 2013)*, pages 365–376, 2013.

M. Reiß and M. Wahl.

Nonasymptotic upper bounds for the reconstruction error of pca.

The Annals of Statistics, 48(2):1098–1123, 2020.

C. Milbradt and M. Wahl.

High-probability bounds for the reconstruction error of pca.

Statistics & Probability Letters, 161:108741, 2020.