GRAN SASSO Science Institute, Intensive Trimester
"Particles, Fluids and Patterns: Analytical and Computational Challenges"
3-4 June 2022

High-dimensional approximation and sampling

Part 3: Manifold approximation

Anthony Nouy

Centrale Nantes, Nantes Université, Laboratoire de Mathématiques Jean Leray

Approximation of a function class

We consider the problem of approximating functions from a subset K of a normed vector space X.

We want to find a low-dimensional space or manifold M_n which approximates well the set K, or a sequence of models with increasing complexity $(M_n)_{n\geq 1}$ that approximate K with a good rate of convergence.

Approximation of a function class

We consider the problem of approximating functions from a subset K of a normed vector space X.

We want to find a low-dimensional space or manifold M_n which approximates well the set K, or a sequence of models with increasing complexity $(M_n)_{n\geq 1}$ that approximate K 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),

Approximation of a function class

We consider the problem of approximating functions from a subset K of a normed vector space X.

We want to find a low-dimensional space or manifold M_n which approximates well the set K, or a sequence of models with increasing complexity $(M_n)_{n\geq 1}$ that approximate K 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,

Approximation of a function class

We consider the problem of approximating functions from a subset K of a normed vector space X.

We want to find a low-dimensional space or manifold M_n which approximates well the set K, or a sequence of models with increasing complexity $(M_n)_{n\geq 1}$ that approximate K 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,

Approximation of a function class

We consider the problem of approximating functions from a subset K of a normed vector space X.

We want to find a low-dimensional space or manifold M_n which approximates well the set K, or a sequence of models with increasing complexity $(M_n)_{n\geq 1}$ that approximate K 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,

Approximation of a function class

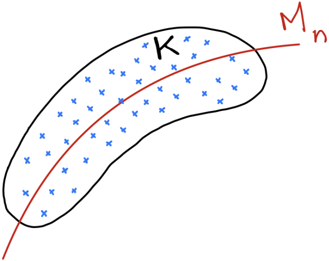
We consider the problem of approximating functions from a subset K of a normed vector space X.

We want to find a low-dimensional space or manifold M_n which approximates well the set K, or a sequence of models with increasing complexity $(M_n)_{n\geq 1}$ that approximate K 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 sets K (application-dependent), M_n can be obtained by manifold approximation (or dimension reduction) methods.

Approximation of a function class

In this part, we present methods for approximating a set K by low-dimensional sets M_n, using samples in K.



A common setting (in statistics) if when K is the range of some vector or function-valued random variable.

Another classical setting is the solution of forward or inverse problems for parameter-dependent equations, where

$$K = \{u(y) : y \in Y\} \quad \text{with} \quad R(u(y); y) = 0.$$

Approximation of a function class

The approximating sets M_n can be

- constructed offline by manifold approximation methods, from samples from K,
- used online to compute approximations of elements in K with low computational complexity, or from limited information.

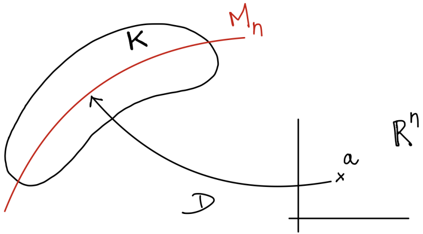
The offline construction may take into account the online approximation process.

Encoder-Decoder

A large class of manifold approximation methods can be described by an encoder $E : K \to \mathbb{R}^n$ and a decoder $D : \mathbb{R}^n \to X$.

The decoder provides a parametrization of a n-dimensional "manifold"

$$M_n = \{D(a) : a \in \mathbb{R}^n\}.$$



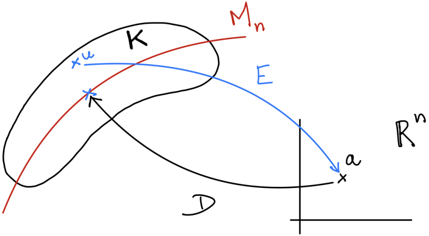
Encoder-Decoder

A large class of manifold approximation methods can be described by an encoder $E : K \to \mathbb{R}^n$ and a decoder $D : \mathbb{R}^n \to X$.

The decoder provides a parametrization of a n-dimensional "manifold"

$$M_n = \{D(a) : a \in \mathbb{R}^n\}.$$

The encoder is related to the approximation process (algorithm). It associates to $u \in K$ a parameter value $a = E(u) \in \mathbb{R}^n$.



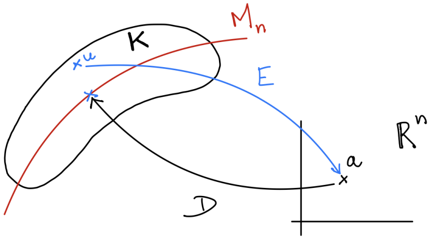
Encoder-Decoder

A large class of manifold approximation methods can be described by an encoder $E : K \to \mathbb{R}^n$ and a decoder $D : \mathbb{R}^n \to X$.

The decoder provides a parametrization of a n-dimensional "manifold"

$$M_n = \{D(a) : a \in \mathbb{R}^n\}.$$

The encoder is related to the approximation process (algorithm). It associates to $u \in K$ a parameter value $a = E(u) \in \mathbb{R}^n$.



An element $u \in K$ is approximated by $D \circ E(u) \in M_n$.

This problem is equivalent to approximating the identity map on K by $D \circ E$ (auto-encoder of K).

Optimal performance

Manifold approximation methods can be classified in terms of the properties of their encoders and decoders.

The optimal performance of a given class $\mathcal{E}_n$ of encoders from X to $\mathbb{R}^n$ and a given class $\mathcal{D}_n$ of decoders from $\mathbb{R}^n \to X$ can be assessed in worst-case setting by

$$\inf_{D\in\mathcal{D}_n, E\in\mathcal{E}_n} \sup_{u\in K} \|u - D\circ E(u)\|_X .$$

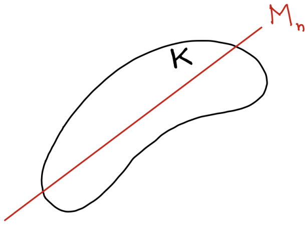
If the set K is equipped with a measure ρ, the optimal performance can be measured in average sense by

$$\inf_{D\in\mathcal{D}_n, E\in\mathcal{E}_n} \left(\int_K \|u - D\circ E(u)\|_X^p d\rho(u) \right)^{1/p} .$$

These errors define measures of complexity (widths) of K.

Linear approximation - Worst case setting

The range M_n of a **linear decoder** $D : \mathbb{R}^n \to X$ is a linear space with dimension at most n.



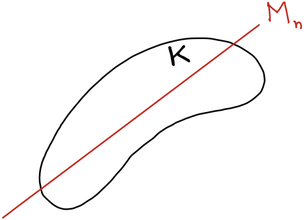
Linear approximation - Worst case setting

The range M_n of a **linear decoder** $D : \mathbb{R}^n \to X$ is a linear space with dimension at most n.

Restricting the **decoder and encoder to be linear** yields the **approximation numbers**

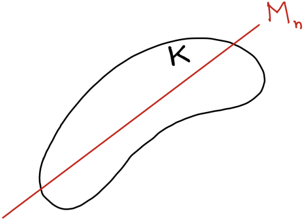
$$a_n(K)_X = \inf_{rank(A)=n} \sup_{u \in K} \|u - Au\|_X,$$

where the infimum is taken over all linear maps $A : X \to X$ with rank n.



Linear approximation - Worst case setting

The range M_n of a **linear decoder** $D : \mathbb{R}^n \to X$ is a linear space with dimension at most n.



Restricting the **decoder and encoder to be linear** yields the **approximation numbers**

$$a_n(K)_X = \inf_{rank(A)=n} \sup_{u\in K} \|u - Au\|_X,$$

where the infimum is taken over all linear maps $A : X \to X$ with rank n.

Restricting **only the decoder to be linear** yields the **Kolmogorov n-width**

$$d_n(K)_X = \inf_{D\in L(\mathbb{R}^n;X)} \sup_{u\in K} \inf_{a\in\mathbb{R}^n} \|u - D(a)\|_X = \inf_{\dim M_n=n} \sup_{u\in K} \inf_{v\in M_n} \|u - v\|_X.$$

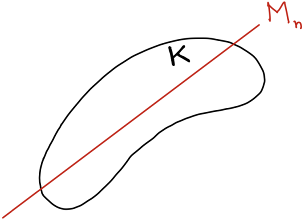
Linear approximation - Worst case setting

The range M_n of a **linear decoder** $D : \mathbb{R}^n \to X$ is a linear space with dimension at most n.

Restricting the **decoder and encoder to be linear** yields the **approximation numbers**

$$a_n(K)_X = \inf_{rank(A)=n} \sup_{u \in K} \|u - Au\|_X,$$

where the infimum is taken over all linear maps $A : X \to X$ with rank n.



Restricting **only the decoder to be linear** yields the **Kolmogorov n-width**

$$d_n(K)_X = \inf_{D \in L(\mathbb{R}^n;X)} \sup_{u \in K} \inf_{a \in \mathbb{R}^n} \|u - D(a)\|_X = \inf_{\dim M_n = n} \sup_{u \in K} \inf_{v \in M_n} \|u - v\|_X.$$

For X a Hilbert space, $a_n(K)_X = d_n(K)_X$ and an optimal auto-encoder $D \circ E$ is given by the orthogonal projection P_{M_n} onto an optimal space M_n.

In practice optimal linear spaces in worst-case error are out of reach but near to optimal spaces M_n can be obtained by greedy algorithms, that generate an increasing sequence of spaces from samples in K [DeVore et al 2013].

Greedy algorithms (for linear approximation)

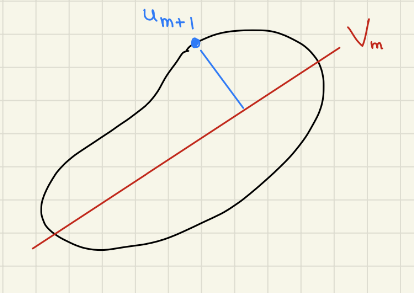
Greedy algorithms can be used for the construction of a hierarchical sequence of spaces $(V_n)_{n\geq 1}$ using samples (snapshots) from K. Spaces are defined by $V_n = span\{u_1, \dots, u_n\}$ where $(u_n)_{i\geq 1}$ is a sequence from K selected greedily.

Greedy algorithms (for linear approximation)

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

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

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



Greedy algorithms (for linear approximation)

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

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

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

Greedy algorithms (for linear approximation)

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

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

In practice, for a computationally feasible algorithm, $E(u(y), V_n)_X$ is replaced by some error estimate $\Delta(u(y), V_n)$, 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

$$\Delta(u(y), V_n) = \|R(u_n(y); y)\|$$

with $u_n(y)$ a Galerkin projection of $u(y)$ onto V_n (Galerkin encoder).

Greedy algorithms (for linear approximation)

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

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

In practice, for a computationally feasible algorithm, $E(u(y), V_n)_X$ is replaced by some error estimate $\Delta(u(y), V_n)$, 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

$$\Delta(u(y), V_n) = \|R(u_n(y); y)\|$$

with $u_n(y)$ a Galerkin projection of $u(y)$ onto V_n (Galerkin encoder).

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_{n+1} satisfying

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

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

$$\sigma_n(K)_X := \sup_{u \in K} E(u, V_n)_X \geq d_n(K)_X$$

Greedy algorithms (for linear approximation)

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

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

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

$$\sigma_n(K)_X := \sup_{u \in K} E(u, V_n)_X \geq d_n(K)_X$$

Assuming $\gamma \geq 1$ is independent of n, 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

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

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

Linear approximation - Average setting

When X is a Hilbert space and the error is measured in average sense, optimal linear decoders are related to the **average Kolmogorov n-width**

$$d_n^{(p)}(K,\rho)_X^p = \inf_{D\in L(\mathbb{R}^n;X)} \int_K \inf_{a\in\mathbb{R}^n} \|u - D(a)\|_X^p d\rho(u) = \inf_{\dim M_n = n} \int_K \|u - P_{M_n}u\|_X^p d\rho(u).$$

For $p=2$ (mean-squared error), an optimal space M_n is given by a dominant eigenspace of the (compact) operator

$$T(v) = \int_K u(u,v)_X d\rho(u), \quad v \in X,$$

and

$$\int_K \|u - P_{M_n}u\|_X^2 d\rho(u) = d_n^{(2)}(K,\rho)_X^2 = \sum_{i>n} \lambda_i(T)$$

where $\{\lambda_i(T)\}_{i\geq 1}$ are the eigenvalues of T, sorted by decreasing value. This is related to the singular value decomposition of the map

$$U : v \in X \mapsto (u,v)_X \in L^2_\rho$$

such that $T = U^*U$.

Linear approximation - Average setting

If ρ is a probability measure, assuming $\bar{u} = \int u \, d\rho(u) = 0$, T is the covariance operator of ρ and M_n is the space of principal components of ρ. This corresponds to Principal Component Analysis (PCA).

If ρ has mean $\bar{u} \neq 0$, we can consider the covariance operator $Tv = \int_K (u - \bar{u})(u - \bar{u}, v)_X \, d\rho(u)$, and $u \in K$ is approximated by $\bar{u} + P_{M_n}(u - \bar{u})$, with M_n the dominant eigenspace of T.

Linear approximation - Average setting

Given samples $u_1, \dots, u_m$ in K, an estimation M_n of the optimal space is given by solving

$$\min_{\dim(M_n)=n} \sum_{i=1}^{m} w_i \|u_i - P_{M_n} u_i\|_X^2$$

that is empirical PCA (up to centering).
The solution is the dominant eigenspace of the operator

$$T_m : v \mapsto \sum_{i=1}^{m} w_i u_i (u_i, v)_X.$$

For an analysis of empirical PCA, see e.g.
[Reiss and Wahl 2020, Milbradt and Wahl 2020].

Linear approximation - Average setting

Assuming $X = \mathbb{R}^N$, M_n is given by the dominant eigenvectors of the matrix

$$\sum_{i=1}^{m} w_i u_i u_i^T \in \mathbb{R}^{N\times N}.$$

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

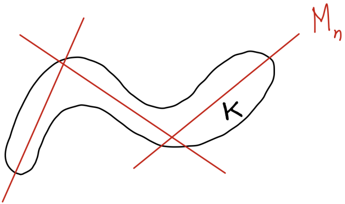
$$A = (u_1, \dots, u_m) \in \mathbb{R}^{N\times m}$$

An important question is how to optimally sample from K ? If ρ is a probability measure, i.i.d. sampling from ρ and $w_i = \frac{1}{m}$ (standard Monte-Carlo) is in general not optimal. Optimal sampling requires an estimation of singular vectors.

Multi-space approximation

M_n can be chosen as a union of N linear spaces of dimension n

$$M_n = \bigcup_{k=1}^{N} V_k$$



A related measure of complexity is the **nonlinear Kolmogorov width** or **library width**

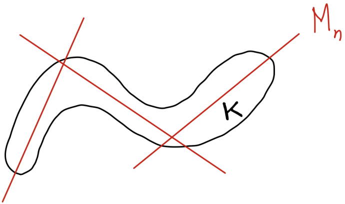
$$d_n(K,N)_X = \inf_{\#\mathcal{L}_{N,n}=N} \sup_{u\in K} \inf_{V\in\mathcal{L}_{N,n}} \|u - P_V u\|_X \qquad \text{[Temlyakov 1998]}$$

where the infimum is taken over all libraries $\mathcal{L}_{N,n}$ of N linear spaces of dimension n.

Multi-space approximation

M_n can be chosen as a union of N linear spaces of dimension n

$$M_n = \bigcup_{k=1}^{N} V_k$$



A related measure of complexity is the **nonlinear Kolmogorov width** or **library width**

$$d_n(K,N)_X = \inf_{\#\mathcal{L}_{N,n}=N} \sup_{u\in K} \inf_{V\in\mathcal{L}_{N,n}} \|u - P_V u\|_X \qquad \text{[Temlyakov 1998]}$$

where the infimum is taken over all libraries $\mathcal{L}_{N,n}$ of N linear spaces of dimension n.

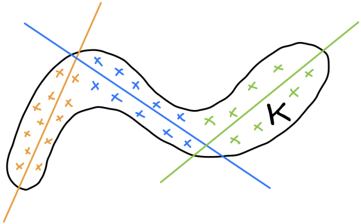
An encoder-decoder view is

$$E : K \to \mathbb{R}^n \times \{1,\dots,N\}, \quad D : \mathbb{R}^n \times \{1,\dots,N\} \to X \quad \text{with} \quad Range(D(\cdot,k)) = V_k$$

An optimal encoding requires a selection of a subspace V_k and a projection onto V_k.

Multi-space approximation

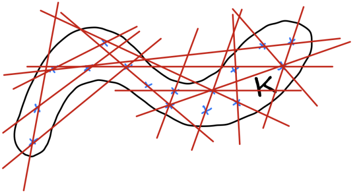
From samples from K, spaces can be obtained by a clustering/partitioning approach (h-p method) [Eftang et al 2010][Bonito et al 2021][Guignard and Mula 2024]



Multi-space approximation

Spaces can be generated from a dictionary $\mathcal{D} = \{u_1, \dots, u_m\}$ of samples from K [Balabanov and Nouy 2021a, Nouy and Pasco 2024]. By considering subspaces with dimension less than n, this yields the model class

$$M_n = \bigcup_{\alpha \in \{1,\dots,m\}^n} V_\alpha, \quad V_\alpha = span\{u_{\alpha_1}, \dots, u_{\alpha_n}\}$$



This is equivalent to n-term approximation

$$M_n = \{\sum_{i=1}^{m} c_i u_i : c \in \mathbb{R}^m, \|c\|_0 \leq n\} := \{D(a,\alpha) = \sum_{i=1}^{n} a_i u_{\alpha_i} : a \in \mathbb{R}^n, \alpha \in \{1, \dots, m\}^n\}.$$

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

We now consider methods for the approximation of a set K by a parametrized nonlinear manifold

$$M_n = \{D(a) : a \in \mathbb{R}^n\}, \quad D : \mathbb{R}^n \to X$$

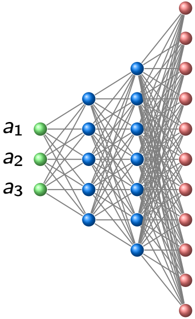
Nonlinear manifold approximation

We now consider methods for the approximation of a set K by a parametrized nonlinear manifold

$$M_n = \{D(a) : a \in \mathbb{R}^n\}, \quad D : \mathbb{R}^n \to X$$

Neural networks are popular tools for this task.

For a finite dimensional space X identified with $\mathbb{R}^N$, a neural network representation can be used for $D : \mathbb{R}^n \to \mathbb{R}^N$.



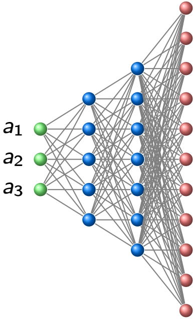
Nonlinear manifold approximation

We now consider methods for the approximation of a set K by a parametrized nonlinear manifold

$$M_n = \{D(a) : a \in \mathbb{R}^n\}, \quad D : \mathbb{R}^n \to X$$

Neural networks are popular tools for this task.

For a finite dimensional space X identified with $\mathbb{R}^N$, a neural network representation can be used for $D : \mathbb{R}^n \to \mathbb{R}^N$.



However, the problem of finding an optimal parameter value a to approximate $u \in K$ (optimal encoding) may be ill-posed or NP-hard.

This suggests to look for a "reasonable" encoder-decoder pair.

Nonlinear manifold approximation

An encoder E and decoder D can be learned from samples in K by minimizing

$$\min_{D,E} \sum_{i=1}^{m} \|u_i - D \circ E(u_i)\|$$

over some restricted classes of maps. Optimal continuous encoder-decoder pairs are related to the notion of **nonlinear manifold width** of [DeVore, Howard and Micchelli 1989]

$$\delta_n(K)_X = \inf_{E \in C(X;\mathbb{R}^n), D \in C(\mathbb{R}^n;X)} \sup_{u \in K} \|u - D \circ E(u)\|_X .$$

Nonlinear manifold approximation

An encoder E and decoder D can be learned from samples in K by minimizing

$$\min_{D,E} \sum_{i=1}^{m} \|u_i - D \circ E(u_i)\|$$

over some restricted classes of maps. Optimal continuous encoder-decoder pairs are related to the notion of **nonlinear manifold width** of [DeVore, Howard and Micchelli 1989]

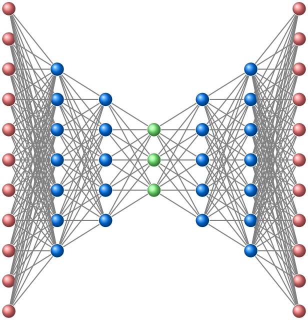
$$\delta_n(K)_X = \inf_{E \in C(X;\mathbb{R}^n), D \in C(\mathbb{R}^n;X)} \sup_{u \in K} \|u - D \circ E(u)\|_X .$$

Further restricting **encoders and decoders to be Lipschitz continuous** yields the notion of **stable manifold width** [Cohen et al 2022].

Lipschitz continuity ensures stability of the approximation process, a crucial property in practice.

Nonlinear manifold approximation

Neural networks can be used for both the encoder $E : \mathbb{R}^N \to \mathbb{R}^n$ and decoder $D : \mathbb{R}^n \to \mathbb{R}^N$.



However, learning D and E is a highly nonlinear optimization problem. Also, controlling the stability is not easy task.

Linear encoder and nonlinear decoder

Restricting the **encoder to be linear and continuous** simplifies its estimation and the control of its stability.

This is related to the notion of **sensing numbers**

$$s_n(K)_X = \inf_{D} \inf_{\ell_{\mathbf{1}},\dots,\ell_n} \sup_{u\in K} \|u - D(\ell_1(u),\dots,\ell_n(u))\|_X$$

where the infimum is taken over all linear forms $\ell_{\mathbf{1}},\dots,\ell_n$ and all nonlinear maps D.

This benchmark is relevant in many applications where the available information $E(u) = (\ell_1(u),\dots,\ell_n(u))$ is linear in u (point evaluations of functions, local averages of functions or more general linear functionals).

Linear/affine encoder and nonlinear decoder

The class of decoders should admit a feasible implementation. A practical approach consists in restricting D to take values in some linear/affine space $\bar{u} + X_N$ with dimension $N \geq n$.

Given a basis $\varphi_1, \dots, \varphi_N$ of X_N, the decoder can be written in the form

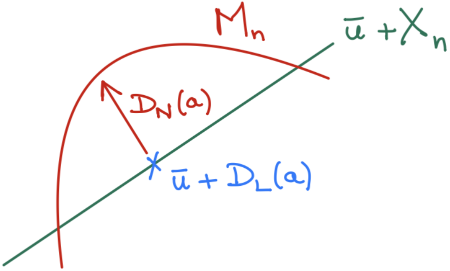
$$D(a) = \bar{u} + \sum_{i=1}^{n} a_i \varphi_i + \sum_{i=n+1}^{N} g_i(a)\varphi_i := \bar{u} + D_L(a) + D_N(a)$$

where the functions $g_i : \mathbb{R}^n \to \mathbb{R}$ are nonlinear maps.

D_L is a linear operator from $\mathbb{R}^n$ to $X_n := span\{\varphi_1, \dots, \varphi_n\}$.

D_N maps $\mathbb{R}^n$ to the complementary space of X_n in X_N.

The range of D is a nonlinear manifold $M_n \subset \bar{u} + X_N$.



Linear/affine encoder and nonlinear decoder

The space X_N and its subspace X_n can be optimized (highly nonlinear problem).

A natural choice is to consider the optimal or near-optimal spaces of linear methods, given by principal component analysis (optimal in mean-squared error) or greedy algorithms (close to optimal in worst case error) [Barnett and Farhat 2022, Geelen et al 2023, Barnett, Farhat and Maday 2023, Geelen et al 2024].

Linear/affine encoder and nonlinear decoder

The structure of the decoder relies on the fact that for

$$u = \bar{u} + \sum_{i=1}^{n} a_i(u)\varphi_i + \sum_{i>n} a_i(u)\varphi_i \in K,$$

the coefficients $a_i(u)$ for $i > n$ may be well approximated as functions $g_i(E(u))$ of the first few coefficients $E(u) = (a_i(u))_{i=1}^{n}$.

Linear/affine encoder and nonlinear decoder

The structure of the decoder relies on the fact that for

$$u = \bar{u} + \sum_{i=1}^{n} a_i(u)\varphi_i + \sum_{i>n} a_i(u)\varphi_i \in K,$$

the coefficients $a_i(u)$ for $i > n$ may be well approximated as functions $g_i(E(u))$ of the first few coefficients $E(u) = (a_i(u))_{i=1}^{n}$.

Natural choices for functions g_i are

- Quadratic polynomials [Barnett and Farhat 2022][Geelen et al 2023]
- Sums of univariate high-order polynomials [Geelen et al 2024]
- Neural networks or random forests [Barnett, Farhat and Maday 2023], [Cohen et al 2023]

Linear/affine encoder and nonlinear decoder

The structure of the decoder relies on the fact that for

$$u = \bar{u} + \sum_{i=1}^{n} a_i(u)\varphi_i + \sum_{i>n} a_i(u)\varphi_i \in K,$$

the coefficients $a_i(u)$ for $i > n$ may be well approximated as functions $g_i(E(u))$ of the first few coefficients $E(u) = (a_i(u))_{i=1}^{n}$.

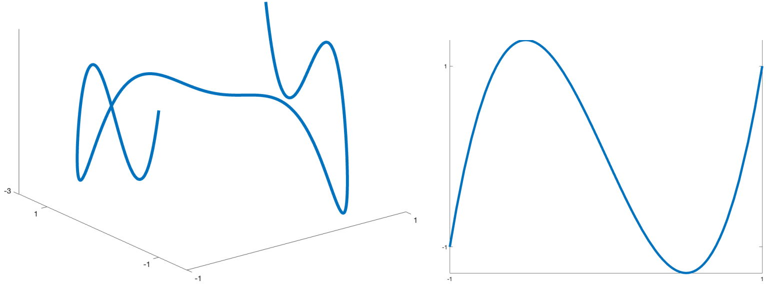
Natural choices for functions g_i are

- Quadratic polynomials [Barnett and Farhat 2022][Geelen et al 2023]
- Sums of univariate high-order polynomials [Geelen et al 2024]
- Neural networks or random forests [Barnett, Farhat and Maday 2023], [Cohen et al 2023]

The relation between $a_i(u)$ and $E(u)$ may be highly nonlinear. Even highly expressive approximation tools may result in poor accuracy, due to the difficulty of learning with limited data.

Linear/affine encoder and nonlinear decoder

In many applications, a coefficient $a_i(u)$ for $i > n$ may have a highly nonlinear relation with the first n coefficients $a = E(u)$ but a much smoother relation when expressed in terms of a and additional coefficients $a_j(u)$ with $n < j < i$.

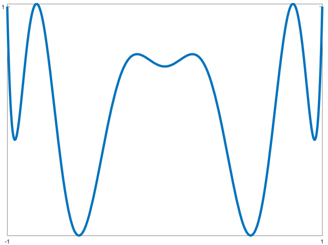


(a) $K = \{(a_1(t), a_2(t), a_3(t)) : t \in [0, 1]\}$

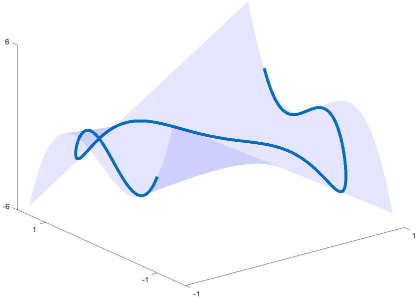
(b) a_2 as function of a_1

Linear/affine encoder and nonlinear decoder

In many applications, a coefficient $a_i(u)$ for $i > n$ may have a highly nonlinear relation with the first n coefficients $a = E(u)$ but a much smoother relation when expressed in terms of a and additional coefficients $a_j(u)$ with $n < j < i$.



(c) a_3 as function of a_1



(d) a_3 as function of (a_1, a_2)

Decoder based on compositional polynomial network

This suggests the following compositional structure of the decoder's functions

$$g_i(a) = f_i(a, (g_j(a))_{n<j\leq n_i}),$$

where the f_i are polynomial functions [Bensalah, Nouy and Soffo 2025].

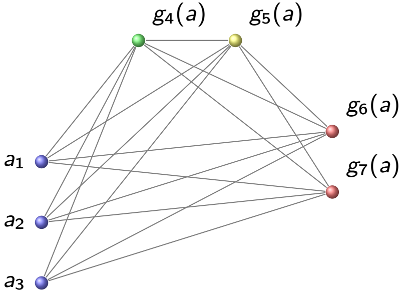


Figure: A compositional polynomial network (CPN) with $N = 7$ and $n = 3$, maximum number of compositions 3.

Decoder based on compositional polynomial network

The variables $(a, (g_j(a))_{n<j\leq n_i})$ take values in a set of measure zero in $\mathbb{R}^{n_i}$, but it is still possible to learn polynomial functions f_i from a limited training set.

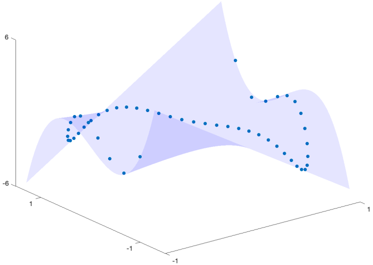


Figure: Learning a_3 as function of (a_1, a_2)

In practice, for high-dimensional approximation of $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ from samples, use of sparse polynomial approximation or tensor networks (low-rank approximation) in $\mathbb{P}_p^{\otimes n_i}$.

Control of error (mean-squared setting)

The mean squared error

$$e_2(D \circ E) := \|id - D \circ E\|_2 := \left(\int_K \|u - D(E(u))\|_X^2 d\rho(u) \right)^{1/2}$$

$$\text{satisfies} \quad e_2(D \circ E)^2 = \sum_{i=n+1}^{N} \epsilon_{i,2}^2 + \|id - P_{X_N}\|_2^2, \quad \epsilon_{i,2} = \|a_i - g_i \circ a\|_2$$

Control of error (mean-squared setting)

The mean squared error

$$e_2(D \circ E) := \|id - D \circ E\|_2 := \left(\int_K \|u - D(E(u))\|_X^2 d\rho(u) \right)^{1/2}$$

$$\text{satisfies} \quad e_2(D \circ E)^2 = \sum_{i=n+1}^{N} \epsilon_{i,2}^2 + \|id - P_{X_N}\|_2^2, \quad \epsilon_{i,2} = \|a_i - g_i \circ a\|_2$$

Given a prescribed precision $0 < \epsilon < 1$, it holds

$$\boxed{e_2(D \circ E) \leq \epsilon e_2(0)}$$

whenever

$$\|id - P_{X_N}\|_2 \leq \beta \epsilon e_2(0) \tag{1}$$

$$\epsilon_{i,2}^2 \leq \bar{\epsilon}_{i,2}^2 := \omega_i (1 - \beta^2) \epsilon^2 e_2(0)^2, \quad \forall i > n \tag{2}$$

where $0 < \beta < 1$ and $(\omega_i)_{i=n+1}^N$ are such that $\sum_{i=n+1}^N \omega_i = 1$.

(1) is achieved by using PCA to define X_N, with a suitable selection of $N = N(\epsilon)$.
(2) is achieved by a control of the approximation of a_i by $f_i((g_j(a))_{j=1}^{n_i})$, using validation.

Control of stability

Given an orthonormal basis $\varphi_1, \dots, \varphi_N$, the encoder $E : X \to \mathbb{R}^n$ is 1-Lipschitz

$$\boxed{\|E(u) - E(u')\|_2 = \|P_{X_n}(u - u')\|_X \leq \|u - u'\|_X}$$

Given $\gamma = (\gamma_i)_{i=n+1}^N$, we equip $\mathbb{R}^{n_i}$ with the norm

$$\|b\|_{i,\gamma} = \max\{\|(b_j)_{j=1}^n\|_2, \max_{n<j\leq n_i} \gamma_j^{-1}|b_j|\}$$

and define the corresponding Lipschitz norm (estimated from samples)

$$\|f_i\|_{i,\gamma} = \max_{b,b'} \frac{|f_i(b) - f_i(b')|}{\|b - b'\|_{i,\gamma}}$$

Given a prescribed Lipschitz constant $L \geq 1$, letting $\|f_i\|_{i,\gamma} = \gamma_i$ and assuming $\gamma_i^2 \leq \bar{\gamma}_i^2$ with $\sum_{i=n+1}^n \bar{\gamma}_i^2 \leq L^2 - 1$, it holds

$$\boxed{\|D(a) - D(a')\|_X \leq L\|a - a'\|_2}$$

Adaptive algorithm

Satisfying the prescribed bounds for precisions

$$\epsilon_{i,p} \leq \bar{\epsilon}_{i,p}$$

or Lipschitz constants

$$\gamma_i = \|f_i\|_{i,\gamma} \leq \bar{\gamma}_i$$

may be a difficult task for some indices $i \in \{1, \dots, N\}$.

This requires to progressively adapt the set of indices $\{1, \dots, n\}$ associated with the encoder.

Prescribed upper bounds $\bar{\epsilon}_{i,p}$ and $\bar{\gamma}_i$ can be updated during the algorithm in order to obtain a sharper control of error and stability.

Numerical illustration: KdV

We consider the Korteweg-de Vries (KdV) equation

$$\frac{\partial u}{\partial t} + 4u\frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0 \quad on \quad [-\pi, \pi] \times [0, 1]$$

with periodic boundary conditions and some initial condition. We consider the manifold

$$K = \{u(\cdot, t) : t \in [0, 1]\}$$

We use 5000 samples $u_i = u(\cdot, t_i)$ as training samples.

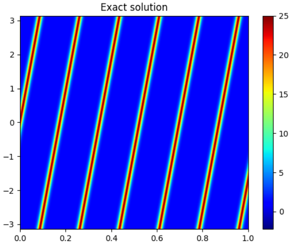


Figure: Function $u(x, t)$.

Method	p	n	N	$\mathbf{RE}_{\text{train}}$	$\mathbf{RE}_{\text{test}}$
Linear	/	5	5	0.435	0.450
Quadratic + AM	2	5	20	0.094	0.099
Univariate + AM	5	5	43	0.081	0.084
Sparse	5	5	43	0.013	0.014
CPN ($\varepsilon = 10^{-4}$)	5	5	43	0.000072	0.000074

Table: Comparison of methods for the same manifold dimension $n = 5$.

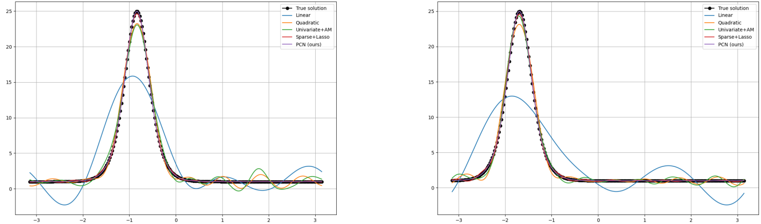
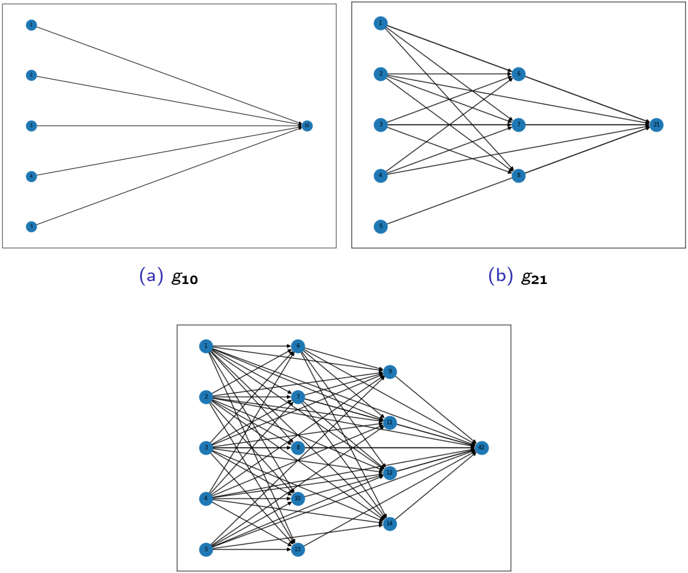


Figure: Comparison of methods for predicting $u(\cdot, t)$ at $t = 0.5$ (left) and $t = 1$ (right). Same dimension $n = 5$.



(a) g_{10}

(b) g_{21}

(c) g_{42}

Figure: Compositional networks for different coefficients

Tolerance	n	N	**Max comp.**	$\mathbf{RE}_{\text{train}}$	$\mathbf{RE}_{\text{test}}$
$\varepsilon = 10^{-1}$	2	15	2	0.062	0.064
$\varepsilon = 10^{-2}$	3	25	5	6.67×10^{-3}	6.84×10^{-3}
$\varepsilon = 10^{-3}$	3	34	8	6.83×10^{-4}	7×10^{-4}
$\varepsilon = 10^{-4}$	5	43	8	7.170×10^{-5}	7.367×10^{-5}
$\varepsilon = 10^{-5}$	6	52	11	6.76×10^{-6}	6.916×10^{-6}
$\varepsilon = 10^{-6}$	11	61	10	7.689×10^{-7}	7.885×10^{-7}

Table: Results of CPN for $p = 5$ and different target precisions ε.

p	n	N	$\mathbf{RE}_{\text{train}}$	$\mathbf{RE}_{\text{test}}$
3	9	43	7.401×10^{-5}	7.574×10^{-5}
4	7	43	7.524×10^{-5}	7.720×10^{-5}
5	5	43	7.170×10^{-5}	7.367×10^{-5}

Table: Results of CPN with different degrees p for $\epsilon = 10^{-4}$

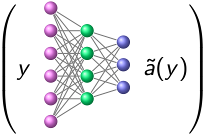
Online approximation

Consider a parametrized manifold

$$K = \{u(y) : y \in Y\}$$

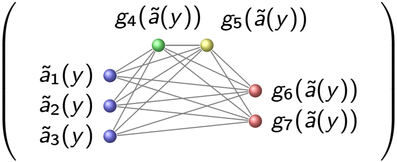
From samples $(y_i, u(y_i))$, we compute offline the decoder D and also an approximation

$$\tilde{a}(y) \approx a(y) := E(u(y))$$



Online, for a parameter value $y \in Y$, we predict

$$\tilde{u}(y) = D(\tilde{a}(y))$$



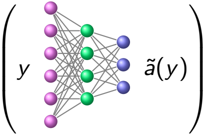
Online approximation

Consider a parametrized manifold

$$K = \{u(y) : y \in Y\}$$

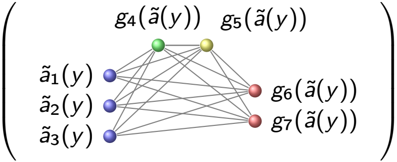
From samples $(y_i, u(y_i))$, we compute offline the decoder D and also an approximation

$$\tilde{a}(y) \approx a(y) := E(u(y))$$



Online, for a parameter value $y \in Y$, we predict

$$\tilde{u}(y) = D(\tilde{a}(y))$$



The error satisfies

$$\|\tilde{u}(y) - u(y)\|_X \leq \|u(y) - D(E(u(y)))\|_X + L\|\tilde{a}(y) - a(y)\|_2.$$

The error $\|\tilde{a}(y) - a(y)\|_2$ should be of the order of $\|u(y) - D(E(u(y)))\|_X / L$.

Online approximation

Even if the constructed manifold $M_n = \{D(a) : a \in \mathbb{R}^n\}$ approximates well K, an approximation/perturbation of the encoding may result in a significant loss of accuracy. Ideally, we would like an element of best approximation

$$\min_{v \in M_n} \|u - v\|_X = \min_{a \in \mathbb{R}^n} \|u - D(a)\|_X$$

This is not achievable using limited information, but a reasonable approximation can be obtained by using advanced learning/optimization strategy (next part).

References I

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.

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.

V. N. Temlyakov.
Nonlinear Kolmogorov widths.
Math. Notes, 63(6):785–795, June 1998.

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.

References II

O. Balabanov and A. Nouy.
Preconditioners for model order reduction by interpolation and random sketching of operators.
arXiv preprint arXiv:2104.12177, 2021.

A. Nouy and A. Pasco.
Dictionary-based model reduction for state estimation.
Adv. Comput. Math., 50(3):1–31, June 2024.

A. Bonito, A. Cohen, R. DeVore, D. Guignard, P. Jantsch, and G. Petrova.
Nonlinear methods for model reduction.
ESAIM: M2AN, 55(2):507–531, Mar. 2021.

D. Guignard and O. Mula.
Tree-Based Nonlinear Reduced Modeling.
In *Multiscale, Nonlinear and Adaptive Approximation II*, pages 267–298. Springer, Cham, Switzerland, Dec. 2024.

J. Barnett and C. Farhat.
Quadratic approximation manifold for mitigating the kolmogorov barrier in nonlinear projection-based model order reduction.
Journal of Computational Physics, 464:111348, Sept. 2022.

References III

J. Barnett, C. Farhat, and Y. Maday.

Neural-network-augmented projection-based model order reduction for mitigating the Kolmogorov barrier to reducibility.

J. Comput. Phys., 492:112420, Nov. 2023.

A. Bensalah, A. Nouy, and J. Soffo.

Nonlinear manifold approximation using compositional polynomial networks.

arXiv e-prints arXiv:2502.05088, Feb. 2025.

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.

A. Cohen, W. Dahmen, O. Mula, and J. Nichols.

Nonlinear Reduced Models for State and Parameter Estimation.

SIAM/ASA Journal on Uncertainty Quantification, Feb. 2022.

R. Geelen, S. Wright, and K. Willcox.

Operator inference for non-intrusive model reduction with quadratic manifolds.

Computer Methods in Applied Mechanics and Engineering, 403:115717, Jan. 2023.

References IV

R. Geelen, L. Balzano, S. Wright, and K. Willcox.
Learning physics-based reduced-order models from data using nonlinear manifolds.
Chaos, 34(3):033122, Mar. 2024.

M. Billaud-Friess, A. Macherey, A. Nouy, and C. Prieur.
A probabilistic reduced basis method for parameter-dependent problems.
In preparation, 2022.

A. Cohen, C. Farhat, Y. Maday, and A. Somacal.
Nonlinear compressive reduced basis approximation for PDE's.
Comptes Rendus. Mécanique, 351(S1):357–374, 2023.

A. Cohen, R. DeVore, G. Petrova, and P. Wojtaszczyk.
Optimal Stable Nonlinear Approximation.
Found. Comput. Math., 22(3):607–648, June 2022.

R. A. DeVore, R. Howard, and C. Micchelli.
Optimal nonlinear approximation.
Manuscripta mathematica, 63(4):469–478, 1989.

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.

References V

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.