A dimension-incremental function approximation method for arbitrary bounded orthonormal product bases

Fabian Taubert

joint work with Lutz Kämmerer and Daniel Potts

Chemnitz University of Technology
Applied Functional Analysis

8th International Conference on Computational Harmonic Analysis
15.09.2022

General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^d \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^d \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^d \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

smooth function f

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

Truncation and approximation

$$S_{\mathrm{I}}^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathrm{I}} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in \mathrm{I}$
- suitable sparse index set $\mathrm{I} \subset \mathbb{Z}^d$

General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^d \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^d \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^d \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

smooth function f

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

Truncation and approximation

$$S_{\mathrm{I}}^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathrm{I}} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \ \forall \boldsymbol{k} \in \mathrm{I}$
- suitable sparse index set $\mathrm{I} \subset \mathbb{Z}^d$

General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^{d} \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^{d} \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^{d} \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

smooth function f

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

Truncation and approximation

$$S_{\mathrm{I}}^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathrm{I}} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in \mathrm{I}$
- suitable sparse index set $\mathrm{I} \subset \mathbb{Z}^d$

General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^d \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^d \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^d \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

smooth function f

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

Truncation and approximation

$$S_{\mathrm{I}}^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathrm{I}} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in \mathrm{I}$
- suitable sparse index set $\mathrm{I} \subset \mathbb{Z}^d$

General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \bigtimes_{j=1}^d \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \bigtimes_{j=1}^d \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^d \mathrm{e}^{2\pi\mathrm{i}k_j x_j}$

smooth function f

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k}\in\mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}$$

Truncation and approximation

$$S_{\mathrm{I}}^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k}\in\mathrm{I}} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}$$

To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in \mathrm{I}$
- suitable sparse index set $\mathrm{I} \subset \mathbb{Z}^d$

General aim for an arbitrary bounded orthonormal product basis (BOPB)

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathcal{D}, \mu)$

by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

Hilbert space $L_2(\mathcal{D}, \mu)$

domain $\mathcal{D} = \times_{j=1}^{d} \mathcal{D}_j \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^{d} \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^{d} \phi_{j,k_j}(x_j)$

smooth function f

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{N}^d} c_{\boldsymbol{k}} \Phi_{\boldsymbol{k}}(\boldsymbol{x})$$

Truncation and approximation

$$S_{\mathrm{I}}^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathrm{I}} \hat{f}_{\boldsymbol{k}} \Phi_{\boldsymbol{k}}(\boldsymbol{x})$$

To compute

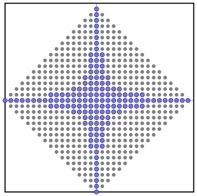
- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \ \forall \boldsymbol{k} \in \mathrm{I}$
- suitable sparse index set $\mathrm{I} \subset \mathbb{N}^d$

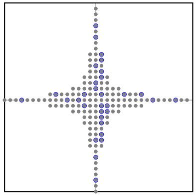
Problem: How to find a good, s-sparse index set I?

- first idea:
  - choose a large search space $\Gamma \supset \mathrm{I}, |\Gamma| \gg |\mathrm{I}|$
  - compute all coefficients
  - choose indices corresponding to the s largest coefficients
- but: unfeasible in practice for large Γ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

Problem: How to find a good, s-sparse index set I?

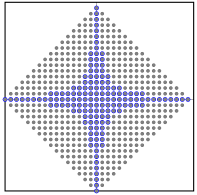
- first idea:
  - choose a large search space $\Gamma \supset \mathrm{I}, |\Gamma| \gg |\mathrm{I}|$
  - compute all coefficients
  - choose indices corresponding to the s largest coefficients
- but: unfeasible in practice for large Γ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

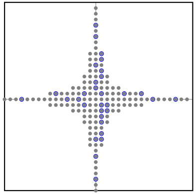




Problem: How to find a good, s-sparse index set I?

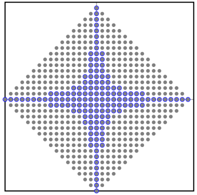
- first idea:
  - choose a large search space $\Gamma \supset \mathrm{I}, |\Gamma| \gg |\mathrm{I}|$
  - compute all coefficients
  - choose indices corresponding to the s largest coefficients
- but: unfeasible in practice for large Γ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

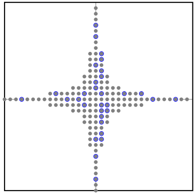




Problem: How to find a good, s-sparse index set I?

- first idea:
    - choose a large search space $\Gamma \supset \mathrm{I}, |\Gamma| \gg |\mathrm{I}|$
    - compute all coefficients
    - choose indices corresponding to the s largest coefficients
- but: unfeasible in practice for large Γ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

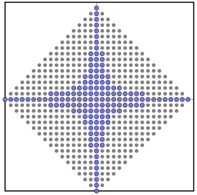


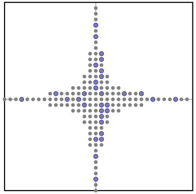


Problem: How to find a good, s-sparse index set I?

- first idea:
  - choose a large search space $\Gamma \supset \mathrm{I}, |\Gamma| \gg |\mathrm{I}|$
  - compute all coefficients
  - choose indices corresponding to the s largest coefficients
- but: unfeasible in practice for large Γ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

Question: Can we ensure to detect all important indices?

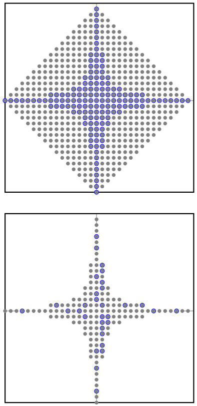




Problem: How to find a good, s-sparse index set I?

- first idea:
  - choose a large search space $\Gamma \supset \mathrm{I}, |\Gamma| \gg |\mathrm{I}|$
  - compute all coefficients
  - choose indices corresponding to the s largest coefficients
- but: unfeasible in practice for large Γ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

Question: Can we ensure to detect all important indices? $\rightarrow$ YES!



Previous works on sparse, high-dimensional approximation:

- our dimension-incremental approach in the Fourier setting:
  - sparse FFT using rank-1 lattices
    [Potts, Volkmer '16], [Kämmerer, Potts, Volkmer '21], [Kämmerer, Krahmer, Volkmer '22]
  - application to differential equations with high-dimensional random coefficients
    [Bochmann, Kämmerer, Potts '20], [Kämmerer, Potts, T. '22]
- other approximation methods in the Fourier setting:
  e.g. [Iwen '13], [Indyk, Kapralov '14], [Choi, Christlieb, Wang '19], ...
- sparse high-dimensional approximation in more general bases:
  - sparse polynomial chaos expansions
    literature survey: [Lüthen, Marelli, Sudret '21]
  - compressive sensing approach for BOPB
    [Choi, Iwen, Krahmer '20], [Choi, Iwen, Volkmer '21]

Previous works on sparse, high-dimensional approximation:

- our dimension-incremental approach in the Fourier setting:
  - sparse FFT using rank-1 lattices
    [Potts, Volkmer '16], [Kämmerer, Potts, Volkmer '21], [Kämmerer, Krahmer, Volkmer '22]
  - application to differential equations with high-dimensional random coefficients
    [Bochmann, Kämmerer, Potts '20], [Kämmerer, Potts, T. '22]
- other approximation methods in the Fourier setting:
  e.g. [Iwen '13], [Indyk, Kapralov '14], [Choi, Christlieb, Wang '19], ...
- sparse high-dimensional approximation in more general bases:
  - sparse polynomial chaos expansions
    literature survey: [Lüthen, Marelli, Sudret '21]
  - compressive sensing approach for BOPB
    [Choi, Iwen, Krahmer '20], [Choi, Iwen, Volkmer '21]

Previous works on sparse, high-dimensional approximation:

- our dimension-incremental approach in the Fourier setting:
  - sparse FFT using rank-1 lattices
    [Potts, Volkmer '16], [Kämmerer, Potts, Volkmer '21], [Kämmerer, Krahmer, Volkmer '22]
  - application to differential equations with high-dimensional random coefficients
    [Bochmann, Kämmerer, Potts '20], [Kämmerer, Potts, T. '22]
- other approximation methods in the Fourier setting:
  e.g. [Iwen '13], [Indyk, Kapralov '14], [Choi, Christlieb, Wang '19], ...
- sparse high-dimensional approximation in more general bases:
  - sparse polynomial chaos expansions
    literature survey: [Lüthen, Marelli, Sudret '21]
  - compressive sensing approach for BOPB
    [Choi, Iwen, Krahmer '20], [Choi, Iwen, Volkmer '21]

Projected coefficients for the dimensions $\{1, \dots, t\}$

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) \coloneqq \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi}, \tilde{\boldsymbol{x}})\, \overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\, \mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi}) \qquad \forall \tilde{\boldsymbol{x}} \in \mathcal{D}_{\{t+1,\dots,d\}}$$

Projected coefficients are an indicator for the importance of the indices $(\boldsymbol{k}, \boldsymbol{h})$ with arbitrary $\boldsymbol{h}$, since

$$c_{\mathfrak{u},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) = \dots = \sum_{\boldsymbol{h}\in\mathbb{N}^{d-t}} c_{(\boldsymbol{k},\boldsymbol{h})} \Phi_{\{t+1,\dots,d\},(\boldsymbol{k},\boldsymbol{h})}(\tilde{\boldsymbol{x}}).$$

Example: Fourier setting with $d = 3$ and $t = 2$

$$c_{\{1,2\},\boldsymbol{k}}(\tilde{x}) \coloneqq \int_{\mathbb{T}^2} f(\boldsymbol{\xi}, \tilde{x})\, \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\xi}}\, \mathrm{d}\boldsymbol{\xi} = \sum_{h\in\mathbb{Z}} c_{(\boldsymbol{k},h)} \mathrm{e}^{2\pi\mathrm{i}h\tilde{x}} \qquad \forall \tilde{x} \in \mathbb{T}$$

$\rightarrow$ Works analogously for single dimensions $\{t\}$ and arbitrary $\mathfrak{u} \subset \{1, \dots, d\}$.

Projected coefficients for the dimensions $\{1, \dots, t\}$

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi}, \tilde{\boldsymbol{x}})\, \overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\, \mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi}) \qquad \forall \tilde{\boldsymbol{x}} \in \mathcal{D}_{\{t+1,\dots,d\}}$$

Projected coefficients are an indicator for the importance of the indices $(\boldsymbol{k}, \boldsymbol{h})$ with arbitrary $\boldsymbol{h}$, since

$$c_{\mathfrak{u},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) = \dots = \sum_{\boldsymbol{h} \in \mathbb{N}^{d-t}} c_{(\boldsymbol{k},\boldsymbol{h})} \Phi_{\{t+1,\dots,d\},(\boldsymbol{k},\boldsymbol{h})}(\tilde{\boldsymbol{x}}).$$

Example: Fourier setting with $d = 3$ and $t = 2$

$$c_{\{1,2\},\boldsymbol{k}}(\tilde{x}) := \int_{\mathbb{T}^2} f(\boldsymbol{\xi}, \tilde{x})\, \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\xi}}\, \mathrm{d}\boldsymbol{\xi} = \sum_{h \in \mathbb{Z}} c_{(\boldsymbol{k},h)} \mathrm{e}^{2\pi\mathrm{i}h\tilde{x}} \qquad \forall \tilde{x} \in \mathbb{T}$$

→ Works analogously for single dimensions $\{t\}$ and arbitrary $\mathfrak{u} \subset \{1, \dots, d\}$.

Projected coefficients for the dimensions $\{1,\dots,t\}$

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi}) \qquad \forall\tilde{\boldsymbol{x}}\in\mathcal{D}_{\{t+1,\dots,d\}}$$

Projected coefficients are an indicator for the importance of the indices $(\boldsymbol{k},\boldsymbol{h})$ with arbitrary $\boldsymbol{h}$, since

$$c_{\mathfrak{u},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) = \ldots = \sum_{\boldsymbol{h}\in\mathbb{N}^{d-t}} c_{(\boldsymbol{k},\boldsymbol{h})}\,\Phi_{\{t+1,\dots,d\},(\boldsymbol{k},\boldsymbol{h})}(\tilde{\boldsymbol{x}}).$$

Example: Fourier setting with $d=3$ and $t=2$

$$c_{\{1,2\},\boldsymbol{k}}(\tilde{x}) := \int_{\mathbb{T}^2} f(\boldsymbol{\xi},\tilde{x})\,\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\xi}}\,\mathrm{d}\boldsymbol{\xi} = \sum_{h\in\mathbb{Z}} c_{(\boldsymbol{k},h)}\mathrm{e}^{2\pi\mathrm{i}h\tilde{x}} \qquad \forall\tilde{x}\in\mathbb{T}$$

$\rightarrow$ Works analogously for single dimensions $\{t\}$ and arbitrary $\mathfrak{u}\subset\{1,\dots,d\}$.

Projected coefficients for the dimensions $\{1, \dots, t\}$

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi}, \tilde{\boldsymbol{x}}) \, \overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})} \, \mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi}) \qquad \forall \tilde{\boldsymbol{x}} \in \mathcal{D}_{\{t+1,\dots,d\}}$$

Projected coefficients are an indicator for the importance of the indices $(\boldsymbol{k}, \boldsymbol{h})$ with arbitrary $\boldsymbol{h}$, since

$$c_{\mathfrak{u},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) = \dots = \sum_{\boldsymbol{h} \in \mathbb{N}^{d-t}} c_{(\boldsymbol{k},\boldsymbol{h})} \Phi_{\{t+1,\dots,d\},(\boldsymbol{k},\boldsymbol{h})}(\tilde{\boldsymbol{x}}).$$

Example: Fourier setting with $d = 3$ and $t = 2$

$$c_{\{1,2\},\boldsymbol{k}}(\tilde{x}) := \int_{\mathbb{T}^2} f(\boldsymbol{\xi}, \tilde{x}) \, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{\xi}} \, \mathrm{d}\boldsymbol{\xi} = \sum_{h \in \mathbb{Z}} c_{(\boldsymbol{k},h)} \mathrm{e}^{2\pi \mathrm{i} h \tilde{x}} \qquad \forall \tilde{x} \in \mathbb{T}$$

$\rightarrow$ Works analogously for single dimensions $\{t\}$ and arbitrary $\mathfrak{u} \subset \{1, \dots, d\}$.

Key idea of the dimension-incremental algorithm

- Construct a candidate set with possible indices in the current dimension(s).
- Compute approximations of projected coefficients: use a reconstruction algorithm (e.g. via a cubature formula).
- Build the index set: keep indices with large projected coefficients and neglect indices with small projected coefficients.

→ Increase the dimension by combining different detected index sets of smaller dimension.

Example for a kind of reconstruction methods: cubature formulas (nodes ξ_j, weights w_j)

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi})$$
$$\approx \sum_{j=1}^{M} w_j f(\boldsymbol{\xi}_j,\tilde{\boldsymbol{x}})\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi}_j)} =: \hat{f}_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}})$$

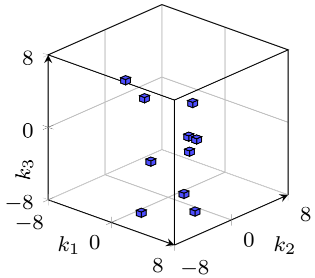
Key idea of the dimension-incremental algorithm

- Construct a candidate set with possible indices in the current dimension(s).
- Compute approximations of projected coefficients: use a reconstruction algorithm (e.g. via a cubature formula).
- Build the index set: keep indices with large projected coefficients and neglect indices with small projected coefficients.

→ Increase the dimension by combining different detected index sets of smaller dimension.

Example for a kind of reconstruction methods: cubature formulas (nodes $\boldsymbol{\xi}_j$, weights w_j)

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi})$$
$$\approx \sum_{j=1}^{M} w_j f(\boldsymbol{\xi}_j,\tilde{\boldsymbol{x}})\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi}_j)} =: \hat{f}_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}})$$

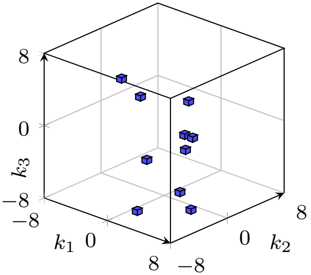


$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \dots, 8\}^3$

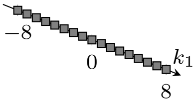
index candidates

- choose candidates according to search space Γ
- construct sampling nodes using a random anchor $\tilde{\boldsymbol{x}} = (\tilde{x}_2, \tilde{x}_3)$
- compute approx. proj. coef. $\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})$
- choose s indices by thresholding $|\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})| \geq \delta_+$

sampling nodes $\{(\frac{\ell}{17}, \tilde{x}_2, \tilde{x}_3)\}_{\ell=0}^{16}$



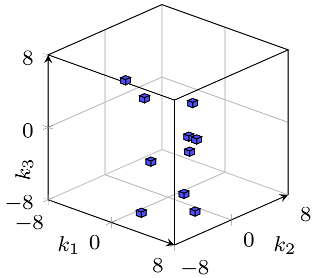
$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$



index candidates

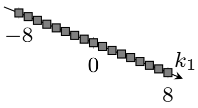
- choose candidates according to search space Γ
- construct sampling nodes using a random anchor $\tilde{\boldsymbol{x}} = (\tilde{x}_2, \tilde{x}_3)$
- compute approx. proj. coef. $\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})$
- choose s indices by thresholding $|\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})| \geq \delta_+$

sampling nodes $\{(\frac{\ell}{17}, \tilde{x}_2, \tilde{x}_3)\}_{\ell=0}^{16}$



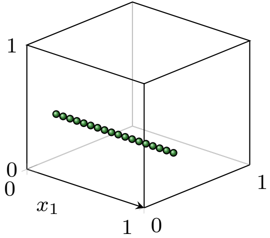
$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$

- choose candidates according to search space Γ
- construct sampling nodes using a random anchor $\tilde{\boldsymbol{x}} = (\tilde{x}_2, \tilde{x}_3)$
- compute approx. proj. coef. $\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})$
- choose s indices by thresholding $|\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})| \geq \delta_+$

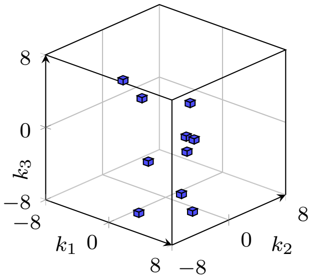


index candidates

construct $\longrightarrow$ sampling set

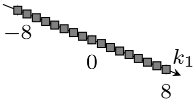


sampling nodes $\{(\frac{\ell}{17}, \tilde{x}_2, \tilde{x}_3)\}_{\ell=0}^{16}$



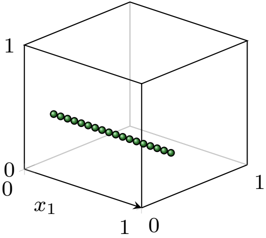
$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$

- choose candidates according to search space Γ
- construct sampling nodes using a random anchor $\tilde{\boldsymbol{x}} = (\tilde{x}_2, \tilde{x}_3)$
- compute approx. proj. coef. $\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})$
- choose s indices by thresholding $|\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})| \geq \delta_+$

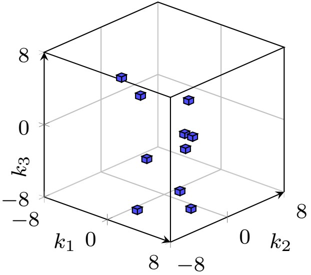


index candidates

reconstruction
⟵
algorithm

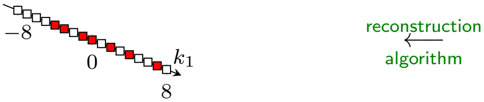


sampling nodes $\{(\frac{\ell}{17}, \tilde{x}_2, \tilde{x}_3)\}_{\ell=0}^{16}$

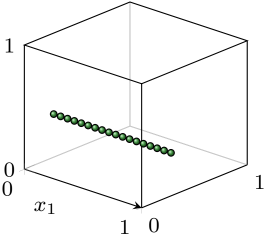


$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$

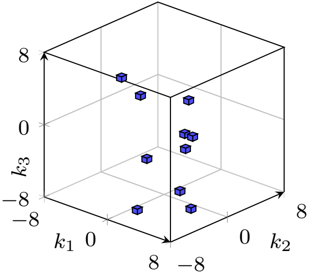
- choose candidates according to search space Γ
- construct sampling nodes using a random anchor $\tilde{x} = (\tilde{x}_2, \tilde{x}_3)$
- compute approx. proj. coef. $\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})$
- choose s indices by thresholding $|\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})| \geq \delta_+$



detected indices $\mathrm{I}^{(1)}$

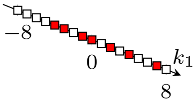


sampling nodes $\{(\frac{\ell}{17}, \tilde{x}_2, \tilde{x}_3)\}_{\ell=0}^{16}$



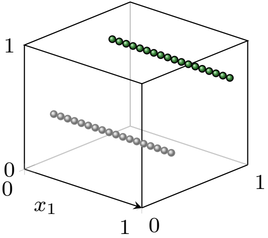
$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$

- choose candidates according to search space Γ
- construct sampling nodes using a random anchor $\tilde{\boldsymbol{x}} = (\tilde{x}_2, \tilde{x}_3)$
- compute approx. proj. coef. $\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})$
- choose s indices by thresholding $|\hat{f}_{\{1\},k_1}(\tilde{\boldsymbol{x}})| \geq \delta_+$



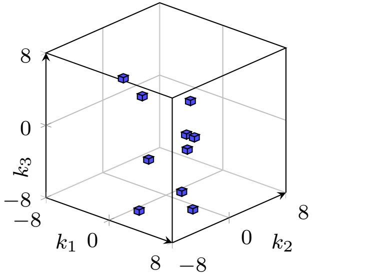
detected indices $\mathrm{I}^{(1)}$

reconstruction ⟵ algorithm

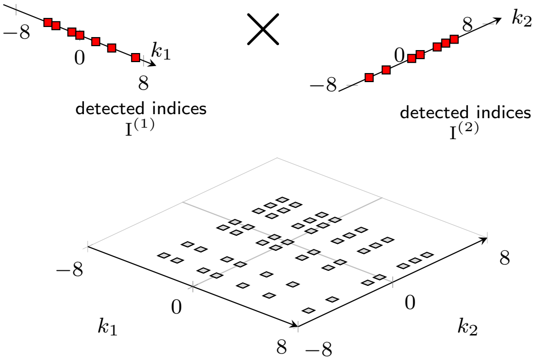


sampling nodes $\{(\frac{\ell}{17}, \tilde{x}_2, \tilde{x}_3)\}_{\ell=0}^{16}$

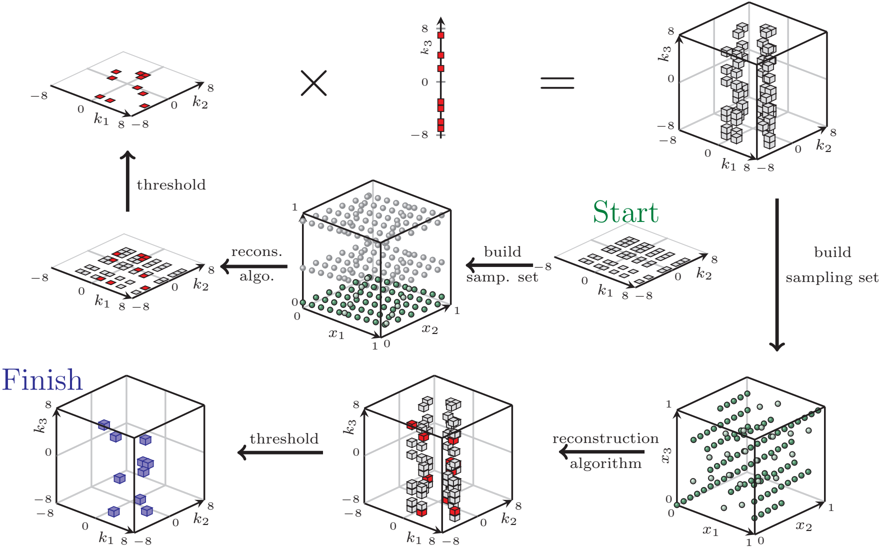
+ repeat with other $\tilde{\boldsymbol{x}}$ (r detection iterations)



$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$



index candidates $\mathrm{I}^{(1)} \times \mathrm{I}^{(2)}$


Start
build
sampling set
reconstruction
algorithm
threshold
Finish
build
samp. set
recons.
algo.
threshold

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant B, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $\mathrm{I}_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space Γ and sparsity s: $\mathrm{I}_{3\delta} \subset \Gamma$ and $s \geq |\mathrm{I}_{3\delta}|$
  - detection threshold δ_+: $B(\delta_+ + \delta_\Psi) < 3 \min_{\boldsymbol{k} \in \mathrm{I}_{3\delta}} |c_{\boldsymbol{k}}|$
  - number of detection iterations r:
    $r \geq \left(1 + \frac{3B^3C}{2} |\mathrm{I}_{3\delta}| + \frac{B^3C}{2\delta} \sum_{\boldsymbol{h} \notin \mathrm{I}_{3\delta}} |c_{\boldsymbol{h}}|\right) (\log 3 + \log d + \log |\mathrm{I}_{3\delta}| - \log \varepsilon)$
- **Then, with probability $1 - \varepsilon$, the output index set I of the Algorithm contains $\mathrm{I}_{3\delta}$.**

Improvements: works for **function approximation** in **any** BOPB!

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant B, threshold $\delta > 0$, failure prob. $\varepsilon \in (0, 1)$
- index set: $\mathrm{I}_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space Γ and sparsity s: $\mathrm{I}_{3\delta} \subset \Gamma$ and $s \geq |\mathrm{I}_{3\delta}|$
  - detection threshold δ_+: $B(\delta_+ + \delta_\Psi) < 3 \min_{\boldsymbol{k} \in \mathrm{I}_{3\delta}} |c_{\boldsymbol{k}}|$
  - number of detection iterations r:
    $r \geq \left(1 + \frac{3B^3C}{2} |\mathrm{I}_{3\delta}| + \frac{B^3C}{2\delta} \sum_{\boldsymbol{h} \notin \mathrm{I}_{3\delta}} |c_{\boldsymbol{h}}|\right) (\log 3 + \log d + \log |\mathrm{I}_{3\delta}| - \log \varepsilon)$
- **Then, with probability $1 - \varepsilon$, the output index set I of the Algorithm contains $\mathrm{I}_{3\delta}$.**

Improvements: works for **function approximation** in **any** BOPB!

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant B, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $\mathrm{I}_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space Γ and sparsity s: $\mathrm{I}_{3\delta} \subset \Gamma$ and $s \geq |\mathrm{I}_{3\delta}|$
  - detection threshold δ_+: $B(\delta_+ + \delta_\Psi) < 3 \min_{\boldsymbol{k} \in \mathrm{I}_{3\delta}} |c_{\boldsymbol{k}}|$
  - number of detection iterations r:
    $r \geq \left(1 + \frac{3B^3C}{2} |\mathrm{I}_{3\delta}| + \frac{B^3C}{2\delta} \sum_{\boldsymbol{h} \notin \mathrm{I}_{3\delta}} |c_{\boldsymbol{h}}|\right) (\log 3 + \log d + \log |\mathrm{I}_{3\delta}| - \log \varepsilon)$
- **Then, with probability $1-\varepsilon$, the output index set I of the Algorithm contains $\mathrm{I}_{3\delta}$.**

Improvements: works for **function approximation** in **any** BOPB!

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant B, threshold $\delta > 0$, failure prob. $\varepsilon \in (0, 1)$
- index set: $\mathrm{I}_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space Γ and sparsity s: $\mathrm{I}_{3\delta} \subset \Gamma$ and $s \geq |\mathrm{I}_{3\delta}|$
  - detection threshold δ_+: $B(\delta_+ + \delta_\Psi) < 3 \min_{\boldsymbol{k} \in \mathrm{I}_{3\delta}} |c_{\boldsymbol{k}}|$
  - number of detection iterations r:
    $r \geq \left(1 + \frac{3B^3C}{2} |\mathrm{I}_{3\delta}| + \frac{B^3C}{2\delta} \sum_{\boldsymbol{h} \notin \mathrm{I}_{3\delta}} |c_{\boldsymbol{h}}|\right) (\log 3 + \log d + \log |\mathrm{I}_{3\delta}| - \log \varepsilon)$
- **Then, with probability $1 - \varepsilon$, the output index set I of the Algorithm contains $\mathrm{I}_{3\delta}$.**

Improvements: works for **function approximation** in **any** BOPB!

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D},\mu)$, BOPB constant B, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $\mathrm{I}_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space Γ and sparsity s: $\mathrm{I}_{3\delta} \subset \Gamma$ and $s \geq |\mathrm{I}_{3\delta}|$
  - detection threshold δ_+: $B(\delta_+ + \delta_\Psi) < 3\min_{\boldsymbol{k}\in\mathrm{I}_{3\delta}} |c_{\boldsymbol{k}}|$
  - number of detection iterations r:
    $r \geq \left(1 + \frac{3B^3C}{2}|\mathrm{I}_{3\delta}| + \frac{B^3C}{2\delta}\sum_{\boldsymbol{h}\notin\mathrm{I}_{3\delta}}|c_{\boldsymbol{h}}|\right)(\log 3 + \log d + \log|\mathrm{I}_{3\delta}| - \log\varepsilon)$
- **Then, with probability $1-\varepsilon$, the output index set I of the Algorithm contains $\mathrm{I}_{3\delta}$.**

Improvements: works for **function approximation** in **any** BOPB!

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant B, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $\mathrm{I}_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space Γ and sparsity s: $\mathrm{I}_{3\delta} \subset \Gamma$ and $s \geq |\mathrm{I}_{3\delta}|$
  - detection threshold δ_+: $B(\delta_+ + \delta_\Psi) < 3 \min_{\boldsymbol{k} \in \mathrm{I}_{3\delta}} |c_{\boldsymbol{k}}|$
  - number of detection iterations r:
    $r \geq \left(1 + \frac{3B^3C}{2} |\mathrm{I}_{3\delta}| + \frac{B^3C}{2\delta} \sum_{\boldsymbol{h} \notin \mathrm{I}_{3\delta}} |c_{\boldsymbol{h}}|\right) (\log 3 + \log d + \log |\mathrm{I}_{3\delta}| - \log \varepsilon)$
- **Then, with probability $1 - \varepsilon$, the output index set I of the Algorithm contains $\mathrm{I}_{3\delta}$.**

Improvements: works for **function approximation** in **any** BOPB!

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant B, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $\mathrm{I}_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space Γ and sparsity s: $\mathrm{I}_{3\delta} \subset \Gamma$ and $s \geq |\mathrm{I}_{3\delta}|$
  - detection threshold δ_+: $B(\delta_+ + \delta_\Psi) < 3 \min_{\boldsymbol{k} \in \mathrm{I}_{3\delta}} |c_{\boldsymbol{k}}|$
  - number of detection iterations r:
    $r \geq \left(1 + \frac{3B^3C}{2} |\mathrm{I}_{3\delta}| + \frac{B^3C}{2\delta} \sum_{\boldsymbol{h} \notin \mathrm{I}_{3\delta}} |c_{\boldsymbol{h}}|\right) (\log 3 + \log d + \log |\mathrm{I}_{3\delta}| - \log \varepsilon)$
- **Then, with probability $1-\varepsilon$, the output index set I of the Algorithm contains $\mathrm{I}_{3\delta}$.**

Improvements: works for **function approximation** in **any** BOPB!

Proof as a blueprint:

- Show, that a single approximated projected coefficient is small (= forgotten) with probability $q < 1$ (due to approximation and random anchor $\tilde{\boldsymbol{x}}$) though it should be large.
- Iteration over r gives us the probability $q^r < 1$ of missing *one* index.
- Apply union bounds to bound the probability of missing *any* index (still depends on q^r).
- Show bound on r such that failure probability for each step becomes smaller than $\varepsilon/(3d)$. Then:

$$\begin{aligned}\mathbb{P}\left(\bigcap_{t=1}^{d} E_{1,t} \cap \bigcap_{t=2}^{d} E_{2,t} \cap \bigcap_{t=2}^{d} E_{3,t}\right) &= 1 - \mathbb{P}\left(\bigcup_{t=1}^{d} E_{1,t}^{\mathrm{c}} \cup \bigcup_{t=2}^{d} E_{2,t}^{\mathrm{c}} \cup \bigcup_{t=2}^{d} E_{3,t}^{\mathrm{c}}\right) \\ &\geq 1 - \sum_{t=1}^{d} \mathbb{P}(E_{1,t}^{\mathrm{c}}) - \sum_{t=2}^{d} \mathbb{P}(E_{2,t}^{\mathrm{c}}) - \sum_{t=2}^{d} \mathbb{P}(E_{3,t}^{\mathrm{c}}) = 1 - \varepsilon.\end{aligned}$$

$E_{1,t}$ = one-dimensional detections work successfully
$E_{2,t}$ = construction of sampling set and reconstruction work successfully
$E_{3,t}$ = higher-dimensional detections work successfully

Test function (from [Potts, Volkmer '16], [Kämmerer, Krahmer, Volkmer '21])

$$f(\boldsymbol{x}) := \prod_{j\in\{1,3,8\}} N_2(x_j) + \prod_{j\in\{2,5,6,10\}} N_4(x_j) + \prod_{j\in\{4,7,9\}} N_6(x_j)$$

$f \in L_2(\mathbb{T}^{10}, \mu)$, with the B-Spline of order $m \in \mathbb{N}$

$$N_m(x) := C_m \sum_{k\in\mathbb{Z}} \operatorname{sinc}\left(\frac{\pi}{m}k\right)^m (-1)^k \mathrm{e}^{2\pi \mathrm{i} kx}$$

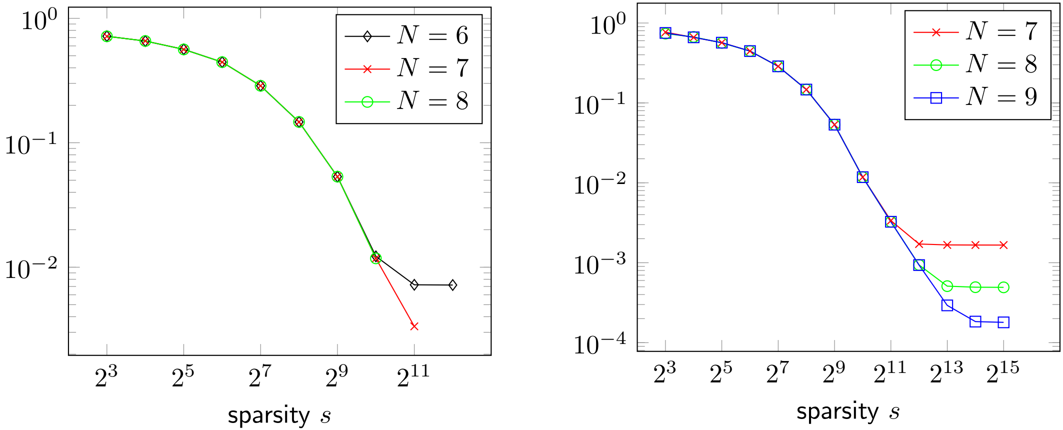
Parameters

- $\Gamma = \left\{\boldsymbol{k} \in \mathbb{Z}^d : \prod_{j=1}^d \max\left(1, 2\,|k_j|\right) \leq 2^N\right\}$
- detection iterations $r = 5$
- detection threshold $\delta_+ = 10^{-12}$

Approximation error

- rel. $L_2(\mathbb{T}^{10}, \mu)$ error:

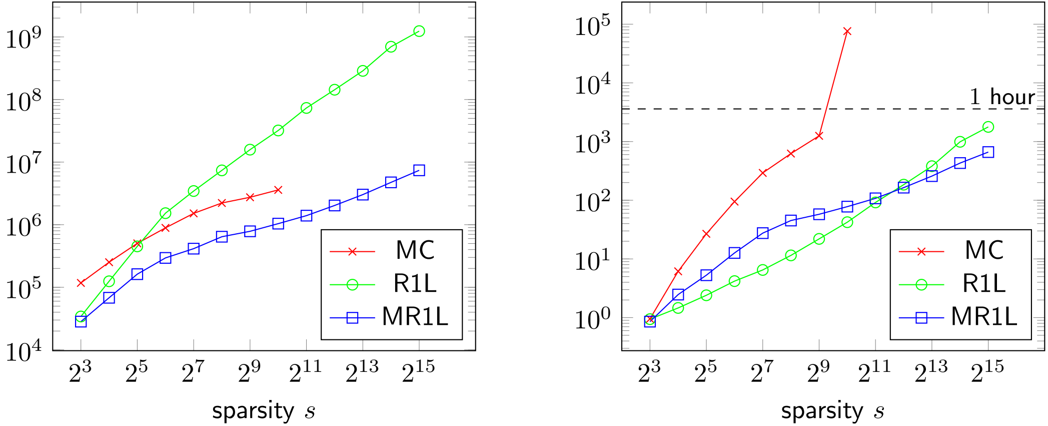
$$\frac{\|f - S_{\mathrm{I}} f\|_{L_2(\mathbb{T}^{10})}}{\|f\|_{L_2(\mathbb{T}^{10})}}$$



(a) approx. error for MC

(b) approx. error for R1L & MR1L

Figure: Approximation results for the 10-dimensional periodic test function



(a) amount of samples for $N = 8$

(b) computation time (in seconds) for $N = 8$

Figure: Approximation results for the 10-dimensional periodic test function

Test function (from [Volkmer '17], [Potts, Volkmer '17])

$$f(\boldsymbol{x}) := \prod_{j\in\{1,3,4,7\}} B_2(x_j) + \prod_{j\in\{2,5,6,8,9\}} B_4(x_j)$$

$f \in L_2([-1,1]^9, \mu_{\text{Cheb}})$, with B_2 and B_4 shifted, scaled and dilated B-Splines of order 2 and 4

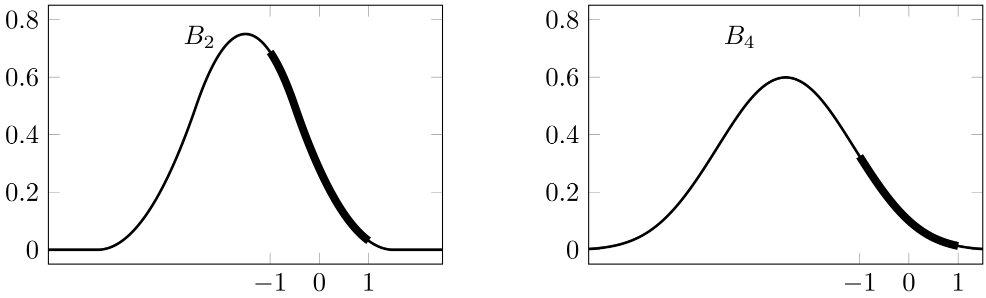
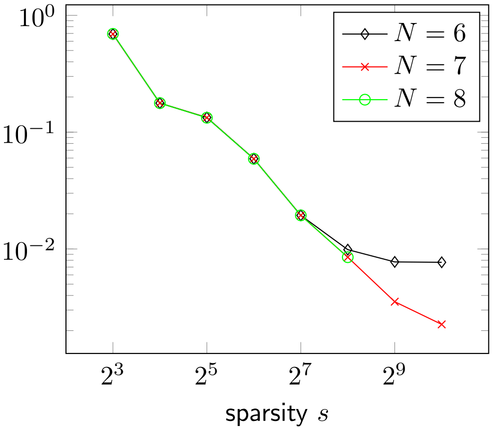
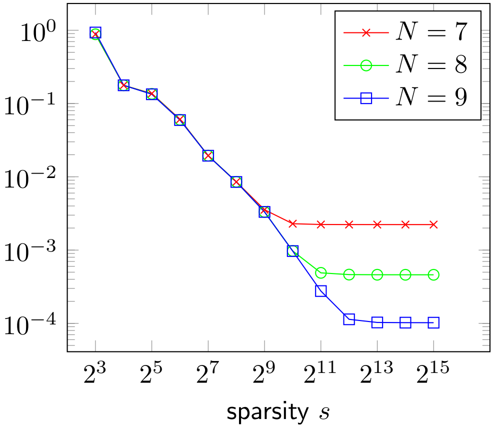


Figure: B-Splines B_2 and B_4 and the considered domain $[-1, 1]$



(a) approx. error for MC



(b) approx. error for CMR1L

Figure: Approximation results for the 9-dimensional non-periodic test function

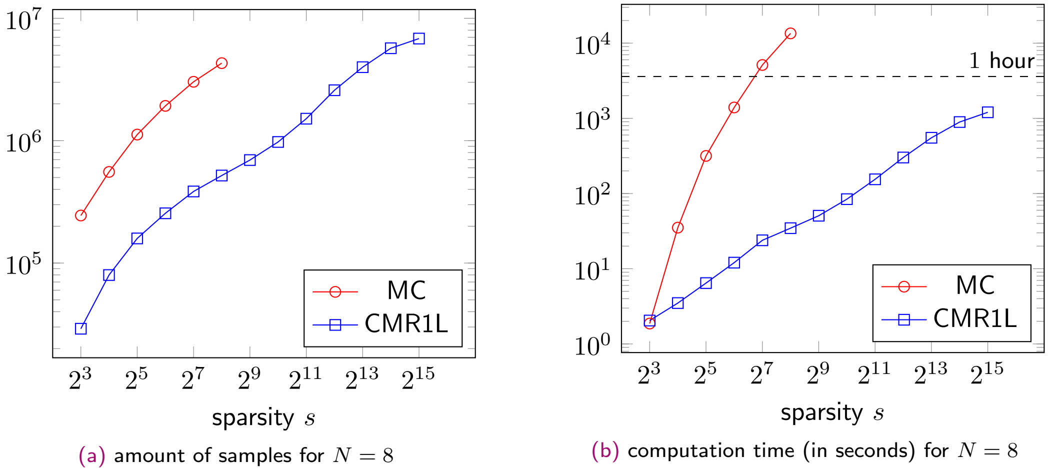


Figure: Approximation results for the 9-dimensional non-periodic test function

- Main achievements of our work:
  - generalization of the dimension-incremental method to arbitrary BOPB
  - first theoretical detection guarantee of the dimension-incremental method for function approximation
  - proof technique can be used as a blueprint when using other reconstruction methods too

- Lutz Kämmerer, Daniel Potts, Fabian Taubert
  A dimension-incremental function approximation method for arbitrary bounded orthonormal product bases
  ArXiv e-prints, 2022 (hopefully next week!)

- Thank you for your attention!

Conclusion

- Main achievements of our work:
  - generalization of the dimension-incremental method to arbitrary BOPB
  - first theoretical detection guarantee of the dimension-incremental method for function approximation
  - proof technique can be used as a blueprint when using other reconstruction methods too

- Lutz Kämmerer, Daniel Potts, Fabian Taubert
  A dimension-incremental function approximation method for arbitrary bounded orthonormal product bases
  ArXiv e-prints, 2022 (hopefully next week!)

- Thank you for your attention!