Model predictivity assessment: incremental test-set selection and accuracy evaluation

ETICS 2022 Research School

Elias Fekhari [1] Bertrand Iooss [1] Joseph Muré [1] Luc Pronzato [2] Maria João Rendas [2]

[1]EDF R&D - 6 quai Watier, Chatou, France

[2]CNRS, Université Côte d'Azur, Laboratoire I3S - 2000 route des lucioles, Sophia Antipolis, France

October 5, 2022

Introduction
Test-set construction methods
- Distance-based design
- Uniformity-based design

Model predictivity estimators
Numerical examples

Machine learning model testing

Machine learning model (or metamodel)

$\eta_m : \mathbb{R}^d \to \mathbb{R}$ built on a given learning set $(\mathbf{X}_m, \mathbf{y}_m)$,
surrogate of the true model $y : \mathbb{R}^d \to \mathbb{R}$

Learning set

$\mathbf{y}_m = [y(\mathbf{x}^{(1)}), \dots, y(\mathbf{x}^{(m)})]$ are the observed outputs at the points
$\mathbf{X}_m = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\} \subset \mathbb{R}^d$

How to certify its performance?

- which testing protocol should be used?
- which performance metric (or indicator) should be used?

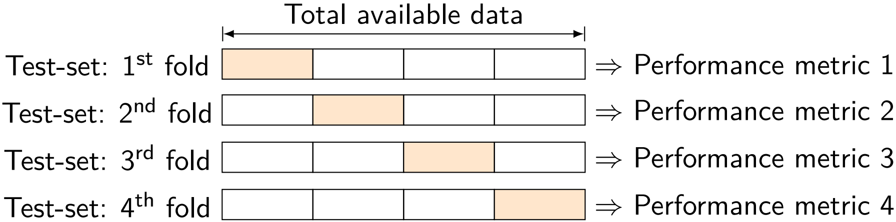
Remarks:

- keep in mind that all we get is as an estimation of its true performance

Classical model testing methods

Cross-validation methods:

k-fold, Leave-One-Out validation (LOO) are the most usual methods[1].



Limits of cross-validation:

- time-consuming ($(n-1)$ models to build for LOO)
- averages the performances of slightly different models: not acceptable for highly sensitive studies (e.g., nuclear industry)

⇒ One solution is to have strictly independent learning and test-set.

How to select an "optimal" test-set?

[1] Tadayoshi Fushiki. "Estimation of prediction error by using K-fold cross-validation". In: *Statistics and Computing* 21.2 (2011), pp. 137–146.

- Introduction
- Test-set construction methods
  - Distance-based design
  - Uniformity-based design
- Model predictivity estimators
- Numerical examples

What is a “good” test-set?

Test-set

$\mathbf{y}_n = [y(\mathbf{x}^{(1)}), \dots, y(\mathbf{x}^{(n)})]$ are the observed outputs at the points $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\} \subset \mathbb{R}^d$

- iterative to ensure a good performance estimation at any size n
- representative of the distribution μ of the input random vector $\mathbf{X}$
- complementary from $\mathbf{X}_m$ to built an enhanced model on the union $\mathbf{X}_{n+m}$

Candidate set

$\mathcal{S}$ is a fairly dense finite subset of $\mathbb{R}^d$ with size $N \gg n$ that quantizes the distribution μ.

Iterative selection

At iteration i, with $\mathbf{X}_i = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(i)}\}$, let us optimize function $\mathcal{A}(\cdot|\mathbf{X}_i)$:

$$\mathbf{x}^{(i+1)} \in \arg\min_{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_i} \mathcal{A}(\mathbf{x}|\mathbf{X}_i) \,. \tag{1}$$

Distance-based design

Geometric construction on a bounded set by sequentially selecting a new point $\mathbf{x}$ as far away as possible from the $\mathbf{x}^{(i)}$ previously selected.

Fully-Sequential Space-Filling[2] (FSSF)

At iteration i, with $\mathbf{X}_i = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(i)}\}$,

$$\mathbf{x}^{(i+1)} \in \underset{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_i}{\arg\max} \left[\min_{j \in \{1,\dots,i\}} \|\mathbf{x} - \mathbf{x}^{(j)}\| \right]. \tag{2}$$

- For non uniform random variables, an iso-probabilistic transform is applied
- FSSF is close to the CADEX algorithm (a.k.a., Coffee house design)

[2] B. Shang and D. Apley. "Fully-sequential space-filling design algorithms for computer experiments". In: *Journal of Quality Technology* 53 (2020), pp. 1–24.

Distance-based design

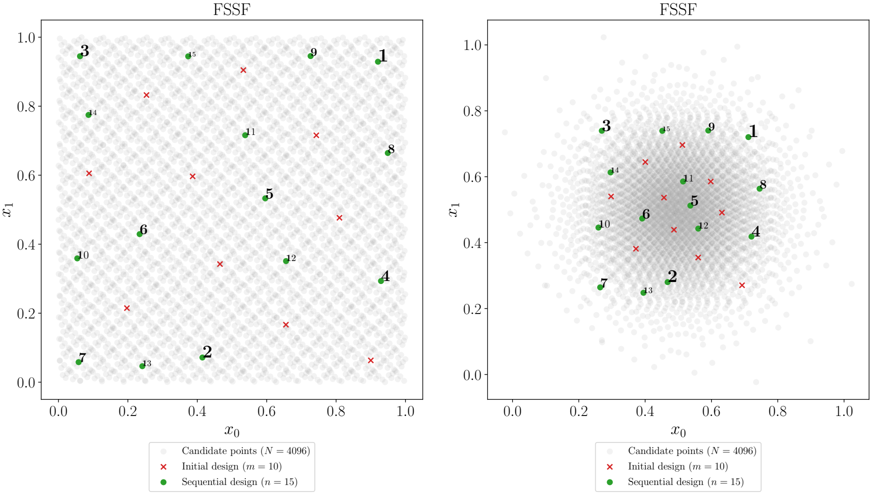


Figure: FSSF sequential test-set designs (uniform and normal 2D)

Maximum Mean Discrepancy[3]

Reproducing Kernel Hilbert Space (RKHS)

For a symmetric and positive definite function $k : \mathcal{X}^2 \to \mathbb{R}$ (kernel).
A RKHS $\mathcal{H}(k)$ is an inner product space of functions $f : \mathcal{X} \to \mathbb{R}$ such that:

- $k(\cdot, \mathbf{x}) \in \mathcal{H}(k), \quad \forall \mathbf{x} \in \mathcal{X}$
- reproducing property $\langle f, k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}(k)} = f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{X}, \forall f \in \mathcal{H}(k)$.

Any positive definite kernel defines a unique RKHS and vice versa.

Maximum Mean Discrepancy (MMD)

The distance between two distributions P and Q:

$$\mathrm{MMD}_k(P, Q) := \sup_{\|f\|_{\mathcal{H}(k)} \leq 1} \left| \int_{\mathcal{X}} f(\mathbf{x}) \mathrm{d}P(\mathbf{x}) - \int_{\mathcal{X}} f(\mathbf{x}) \mathrm{d}Q(\mathbf{x}) \right| \tag{3}$$

A kernel is said to be characteristic when $\mathrm{MMD}(P, Q) = 0 \Leftrightarrow P = Q$.

[3] C.J. Oates. *Minimum Discrepancy Methods in Uncertainty Quantification*. Lecture Notes at ETICS Summer School. 2021.

Maximum Mean Discrepancy

In the following, we consider k as continuous and bounded, according to[4]:

$$\mathrm{MMD}_k(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}(k)} \quad \text{where} \quad \mu_P = \int k(\mathbf{x}, \cdot) dP(\mathbf{x}). \qquad (4)$$

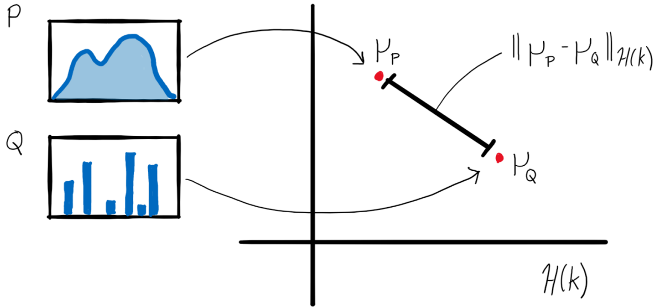


Figure: Kernel mean embedding: mapping distributions in the RKHS $\mathcal{H}(k)$. The distance in the RKHS is the MMD.

[4] Oates, *Minimum Discrepancy Methods in Uncertainty Quantification*.

Uniformity-based design

At iteration n, with $\mathbf{X}_n = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$, the corresponding discrete distribution $\xi_n = \frac{1}{n}\sum_{i=1}^{n} \delta(\mathbf{x}^{(i)})$ and a kernel k:

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_n}{\arg\min} \left(\mathrm{MMD}_k(\mu, \xi_{n+1}(\mathbf{x}))^2 \right) \tag{5}$$

Kernel herding[5]

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_n}{\arg\min} \left(\frac{1}{n} \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}^{(i)}) - \frac{1}{N} \sum_{\mathbf{x}' \in \mathcal{S}} k(\mathbf{x}, \mathbf{x}') \right) \tag{6}$$

Greedy support points[6] (Energy-distance kernel)

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_n}{\arg\min} \left(\frac{1}{N} \sum_{\mathbf{x}' \in \mathcal{S}}^{N} \|\mathbf{x} - \mathbf{x}'\| - \frac{1}{i+1} \sum_{j=1}^{i} \|\mathbf{x} - \mathbf{x}^{(j)}\| \right) \tag{7}$$

[5] Y. Chen, M. Welling, and A. Smola. "Super-samples from kernel herding". In: *Proc. of the 26th UAI Conference*. AUAI Press. 2010.

[6] S. Mak and V.R. Joseph. "Support points". In: *Annals of Statistics* (2018).

Uniformity-based design

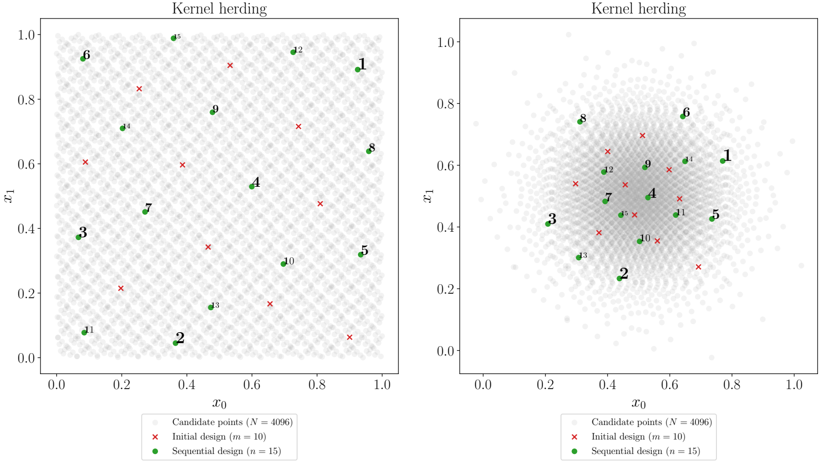


Figure: Kernel herding sequential test-set designs (uniform and normal 2D)

Kernel herding available in pypi package: `otkerneldesign`

Uniformity-based design

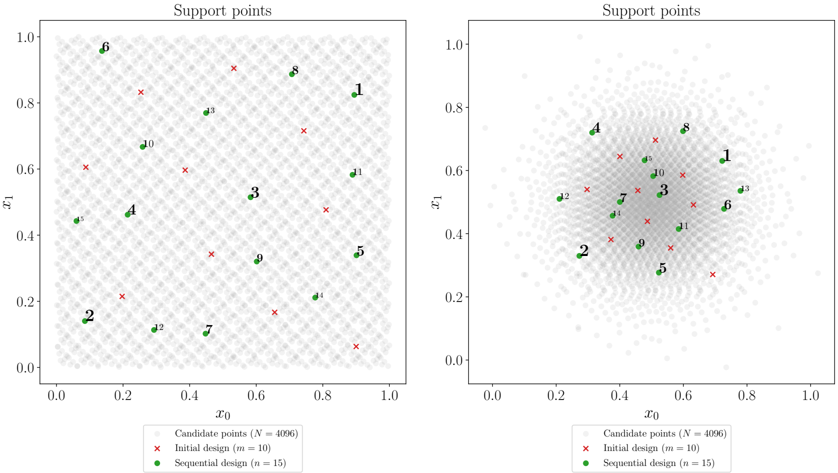


Figure: Greedy support points sequential test-set designs (uniform and normal 2D)

Greedy support points available in pypi package: `otkerneldesign`

Python package documentation

otkerneldesign

Home Doc Examples

Module otkerneldesign

otkerneldesign 0.1.1 documentation » Index of classes » KernelHerding

previous | next | index

KernelHerding

class **otkerneldesign.KernelHerding(*kernel=None, distribution=None, candidate_set_size=None, candidate_set=None, initial_design=None*)**

Incrementally select new design points with kernel herding.

Parameters:

kernel : `openturns.CovarianceModel`
Covariance kernel used to define potentials. By default a product of Matern kernels with smoothness 5/2.

distribution : `openturns.Distribution`
Distribution the design points must represent. If not specified, then *candidate_set* must be specified instead. Even if *candidate_set* is specified, can be useful if it allows the use of analytical formulas.

candidate_set_size : *positive int*
Size of the set of all candidate points. Unnecessary if *candidate_set* is specified. Otherwise, 2^{12} by default.

candidate_set : *2-d list of float*
Large sample that empirically represents a distribution. If not specified, then *distribution* and *candidate_set_size* must be in order to generate it automatically.

initial_design : *2-d list of float*
Sample of points that must be included in the design. Empty by default.

Examples

```
>>> import openturns as ot
>>> import otkerneldesign as otkd
>>> distribution = ot.ComposedDistribution([ot.Normal(0.5, 0.1)] * 2)
>>> dimension = distribution.getDimension()
>>> # Kernel definition
>>> ker_list = [ot.MaternModel([0.1], [1.0], 2.5)] * dimension
>>> kernel = ot.ProductCovarianceModel(ker_list)
>>> # Kernel herding design
>>> kh = otkd.KernelHerding(kernel=kernel, distribution=distribution)
>>> kh_design, _ = kh.select_design(size=20)
```

Previous topic
Index of classes

Next topic
KernelHerdingTensorized

This Page
Show Source

Quick search

Go

- Introduction
- Test-set construction methods
  - Distance-based design
  - Uniformity-based design
- Model predictivity estimators
- Numerical examples

Beyond usual performance metrics

Ideal predictivity coefficient for the predictor η_m

$$Q^2_{\mathrm{ideal}}(\mu) = 1 - \frac{\mathrm{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m)}{\mathrm{Var}_\mu(y(\mathbf{X}))} = 1 - \frac{\int_{\mathcal{X}} [y(\mathbf{x}) - \eta_m(\mathbf{x})]^2 \,\mathrm{d}\mu(\mathbf{x})}{\int_{\mathcal{X}} [y(\mathbf{x}) - \int_{\mathcal{X}} y(\mathbf{x}') d\mu(\mathbf{x}')]^2 \,\mathrm{d}\mu(\mathbf{x})}. \tag{8}$$

Predictivity coefficient: arithmetic estimator

$$\widehat{Q}^2_n = 1 - \frac{\mathrm{ISE}_{\xi_n}(\mathbf{X}_m, \mathbf{y}_m)}{\mathrm{Var}_{\xi_n}(y(\mathbf{X}))} = 1 - \frac{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)}) \right]^2}{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \overline{y}_n \right]^2}. \tag{9}$$

Where $\xi_n = \frac{1}{n} \sum_{i=1}^n \delta(\mathbf{x}^{(i)})\,, \quad \overline{y}_n = \frac{1}{n} \sum_{i=1}^n y(\mathbf{x}^{(i)})$.

- This estimator could exploit the learning set to estimate the variance
- Smart weighting on the ISE could improve the estimation

Beyond usual performance metrics

Assuming the error process $\delta_m(\mathbf{x}) = y(\mathbf{x}) - \eta_m(\mathbf{x}) \sim \mathrm{GP}(0, \sigma^2 K_{|m})$
Let us express the squared error of ISE estimation using ξ_n :

$$\begin{aligned}
\overline{\Delta}^2(\xi_n, \mu; \mathbf{X}_m, \mathbf{y}_m) &= \mathbb{E}\left[(\mathrm{ISE}_{\xi_n}(\mathbf{X}_m, \mathbf{y}_m) - \mathrm{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m))^2\right], \\
&= \mathbb{E}\left[\left(\int_{\mathcal{X}} \delta_m^2(\mathbf{x}) \,\mathrm{d}(\xi_n - \mu)(\mathbf{x})\right)^2\right], \\
&= \sigma^2 \,\mathrm{MMD}^2_{\overline{K}_{|m}}(\xi_n, \mu)\,. \qquad (10)
\end{aligned}$$

Where $\overline{K}_{|m}$ is defined (for an interpolator) as:

$$\overline{K}_{|m}(\mathbf{x}, \mathbf{x}') = 2\,K^2_{|m}(\mathbf{x}, \mathbf{x}') + K_{|m}(\mathbf{x}, \mathbf{x}) K_{|m}(\mathbf{x}', \mathbf{x}')\,,$$

The idea is to find the optimal weights to minimize (10) with a non-uniform measure $\xi_n = \sum_{i=1}^n w_i \delta(\mathbf{x}^{(i)})$. Direct calculation gives:

$$\mathbf{w}_n^* = \overline{\mathbf{K}}_{|m}^{-1}(\mathbf{X}_n) \mathbf{p}_{\overline{K}_{|m},\mu}(\mathbf{X}_n)\,,$$

$$\mathbf{p}_{\overline{K}_{|m},\mu}(\mathbf{X}_n) = \left[\int \overline{K}_{|m}(\mathbf{x}^{(1)}, \mathbf{x}) \,\mathrm{d}\mu(\mathbf{x}), \ldots, \int \overline{K}_{|m}(\mathbf{x}^{(n)}, \mathbf{x}) \,\mathrm{d}\mu(\mathbf{x})\right]^\top$$

Beyond usual performance metrics

Predictivity coefficient: optimally-weighted estimator[7]

$$Q^2_{n*} = 1 - \frac{\sum_{i=1}^{n} w_i^* \left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)}) \right]^2}{\frac{1}{n} \sum_{i=1}^{n} \left[y(\mathbf{x}^{(i)}) - \overline{y}_n \right]^2} . \quad (11)$$

- The weights w_i^* do not depend on the GP variance parameter σ^2
- The denominator could also be weighted

[7] E. Fekhari et al. "Model predictivity assessment: incremental test-set selection and accuracy evaluation". In: *Studies in Theoretical and Applied Statistics, SIS 2021, Pisa, Italy, June 21-25*. Ed. by N. Salvati et al. Springer, to appear, 2022.

Analytical benchmark

Analytical benchmark problems:

- analytical function
- input random variable
- m-size learning set built by optimized LHS (3 sizes corresponding to a poor/good/very good kriging metamodels)
- A reference value for each metamodel computed on a large Monte Carlo test-set

Different test-set sizes, design methods and Q^2 estimators are compared

Analytical benchmark

Analytical benchmark problems:

- analytical function
- input random variable
- m-size learning set built by optimized LHS (3 sizes corresponding to a poor/good/very good kriging metamodels)
- A reference value for each metamodel computed on a large Monte Carlo test-set

Different test-set sizes, design methods and Q^2 estimators are compared

Analytical test-case 3 ("g-sobol" in dimension 8):
The measure μ is uniform on $\mathcal{X} = [0,1]^8$ and $m \in \{15, 30, 100\}$

$$f_3(\mathbf{x}) = \prod_{i=1}^{8} \frac{|4x_i - 2| + a_i}{1 + a_i}, \quad a_i = i^2.$$

Analytical test-cases 1 and 2 (dimension 2) for $\mathbf{x} \in \mathcal{X} = [0, 1]^2$

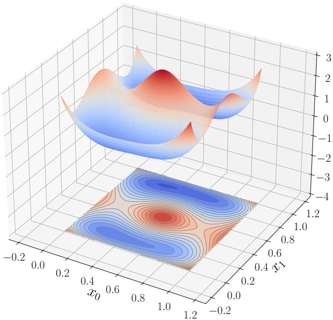


Figure: $f_1(\mathbf{x})$ in test-case 1; μ is uniform; $m \in \{8, 15, 30\}$

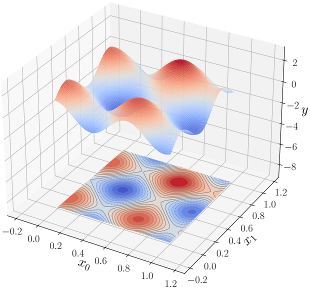


Figure: $f_2(\mathbf{x})$ in test-case 2; μ is standard normal; $m \in \{5, 15, 30\}$

Analytical benchmark results

Analytical test-case 1

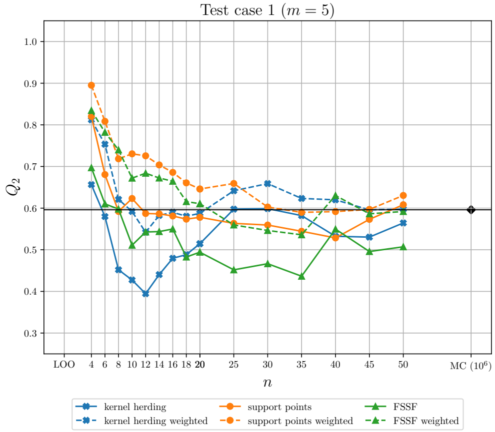


Figure: Predictivity assessment of a poor model with FSSF, SP and KH test sets

Analytical benchmark results

Analytical test-case 1

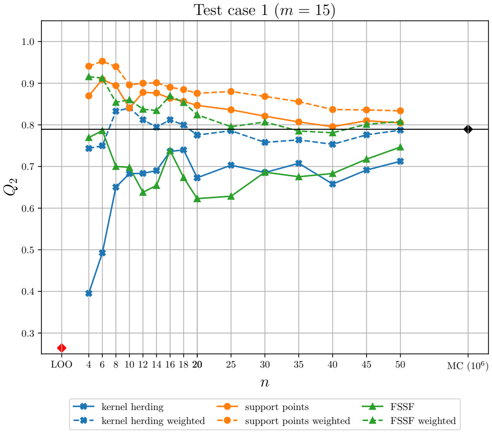


Figure: Predictivity assessment of a good model with FSSF, SP and KH test sets

Analytical benchmark results

Analytical test-case 1

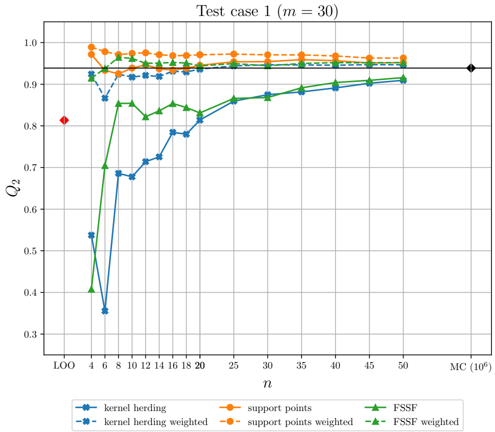


Figure: Predictivity assessment of a very good model with FSSF, SP and KH test sets

Analytical benchmark results

Analysis and interpretation:

- Test-set should at the same time: complement the training set and mimic the target distribution
- Support points and Kernel herding generally perform better
- Kernel herding is sensitive to the chosen kernel
- Each sampling methods are subject to the curse of dimensionality
- Weighting the test-sets helps since it is far from the learning set
- Leave-one-out validation always underestimate, especially for m small
- Once tested, the model can be enhanced by these complementary test-set

Industrial CATHARE use-case

Given data POV:
$\hookrightarrow$ sort decision for each data

Industrial CATHARE use-case

CATHARE test-case:

- Costly numerical simulation code CATHARE2 (20min./run) modeling thermal-hydraulic accident scenario (LOCA-LB) inside nuclear PWR[8]
- 10-dimensional independent random inputs after a screening to reduce the dimension
- Only an existing Monte Carlo dataset $\mathbf{X}_N$ of $N = 10^3$ available
- $\mathbf{X}_N$ includes the test-set $\mathbf{X}_n$ and the complementary training set $\mathbf{X}_{N-n}$

Benchmark protocol:

- Random Cross-Validation (RCV) is repeated ($r = 1000$) to get an empirical distribution of the performance
- To perform the RCV, we use a fast-to-fit Partial Least Squared model

[8] B. Iooss et al. "Numerical studies of the metamodel fitting and validation processes". In: *International Journal of Advances in Systems and Measurements* 3 (2010), pp. 11–21.

Industrial CATHARE use-case results

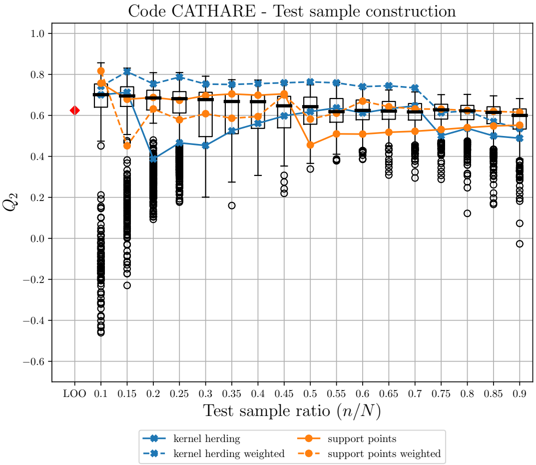


Figure: Estimated Q^2. The box-plots are for random cross-validation, the red diamond (left) is for Q^2_{LOO}.

Industrial CATHARE use-case results

Analysis and interpretation:

- Three behaviours identified (uni or bi-modal empirical distributions)
- Support points seem to have better performances
- Weighted estimator is not as efficient for non-interpolating model
- Good alternative to cross validation for costly to train models

Conclusion

Conclusion and contributions:

- Each method present drawbacks and advantages
- MMD based designs are relevant to select a complementary to the learning set and representative of the target distribution test-set
- A new weighted model performance estimator is proposed and appears to be particularly efficient for interpolators
- This validation is useful when the validation is performed an external part (CV impossible) or if the model training is costly

Perspectives:

✓ Tensorized formulation of the potentials to accelerate the KH

- Non-iterative design leading to complex combinatorial optimization problems

Bibliography

[1] Y. Chen, M. Welling, and A. Smola. "Super-samples from kernel herding". In: *Proc. of the 26th UAI Conference*. AUAI Press. 2010.

[2] E. Fekhari et al. "Model predictivity assessment: incremental test-set selection and accuracy evaluation". In: *Studies in Theoretical and Applied Statistics, SIS 2021, Pisa, Italy, June 21-25*. Ed. by N. Salvati et al. Springer, to appear, 2022.

[3] Tadayoshi Fushiki. "Estimation of prediction error by using K-fold cross-validation". In: *Statistics and Computing* 21.2 (2011), pp. 137–146.

[4] B. Iooss et al. "Numerical studies of the metamodel fitting and validation processes". In: *International Journal of Advances in Systems and Measurements* 3 (2010), pp. 11–21.

[5] S. Mak and V.R. Joseph. "Support points". In: *Annals of Statistics* (2018).

[6] C.J. Oates. *Minimum Discrepancy Methods in Uncertainty Quantification*. Lecture Notes at ETICS Summer School. 2021.

[7] B. Shang and D. Apley. "Fully-sequential space-filling design algorithms for computer experiments". In: *Journal of Quality Technology* 53 (2020), pp. 1–24.

Thank you for your attention

EDF

UNIVERSITÉ
CÔTE D'AZUR

Analytical benchmark results

Analytical test-case 2

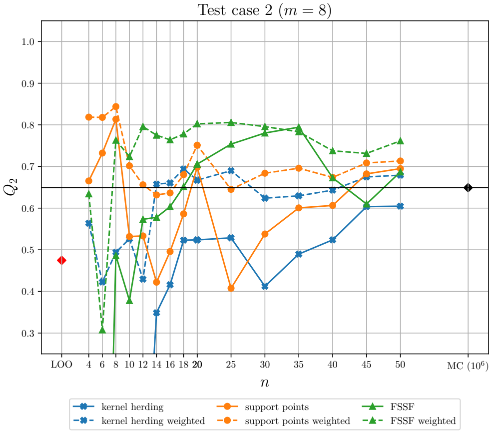


Figure: Predictivity assessment of a poor model with FSSF, SP and KH test sets

Analytical test-case 2

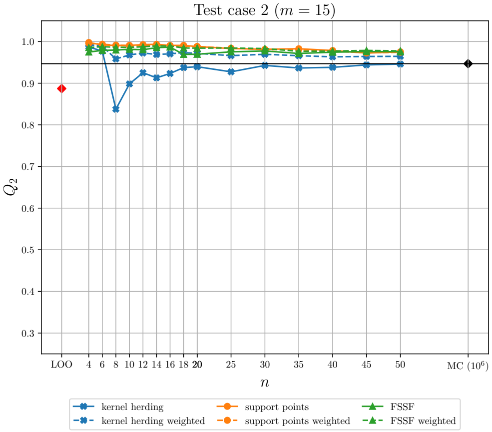


Figure: Predictivity assessment of a good model with FSSF, SP and KH test sets

Analytical benchmark results

Analytical test-case 2

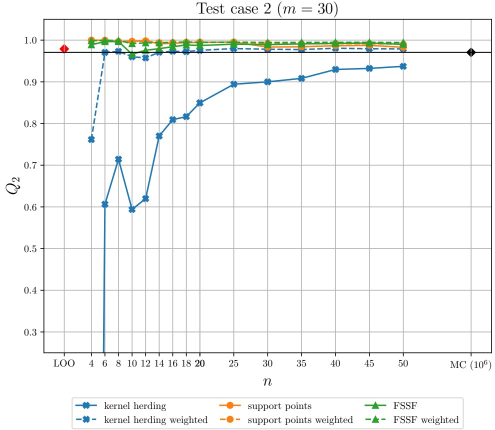


Figure: Predictivity assessment of a very good model with FSSF, SP and KH test sets

Analytical test-case 3

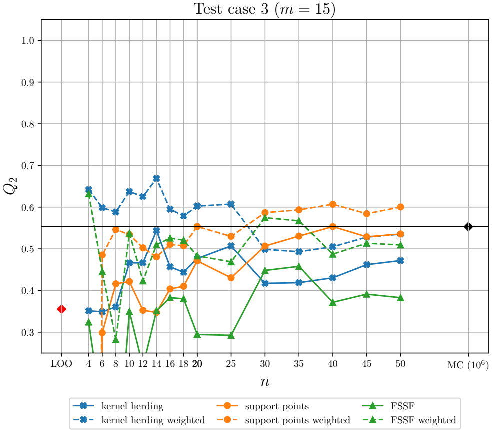


Figure: Predictivity assessment of a poor model with FSSF, SP and KH test sets

Analytical test-case 3

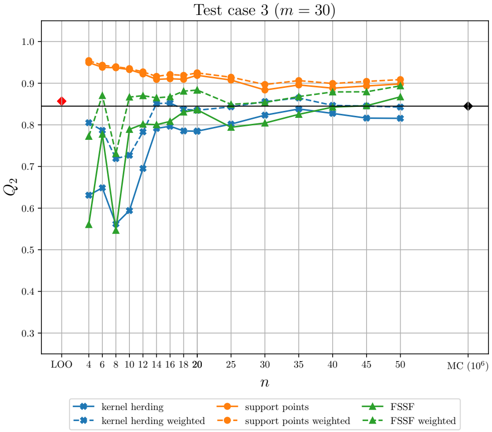


Figure: Predictivity assessment of a good model with FSSF, SP and KH test sets

Analytical test-case 3

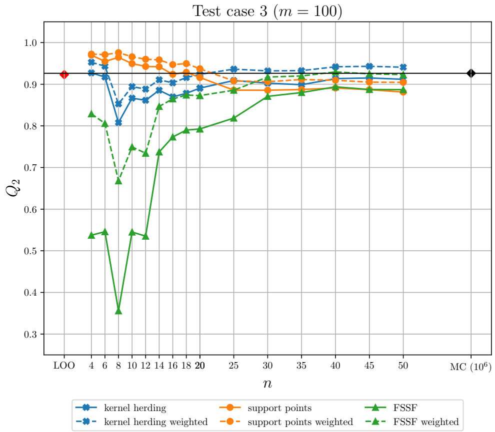


Figure: Predictivity assessment of a very good model with FSSF, SP and KH test sets

Industrial CATHARE use-case

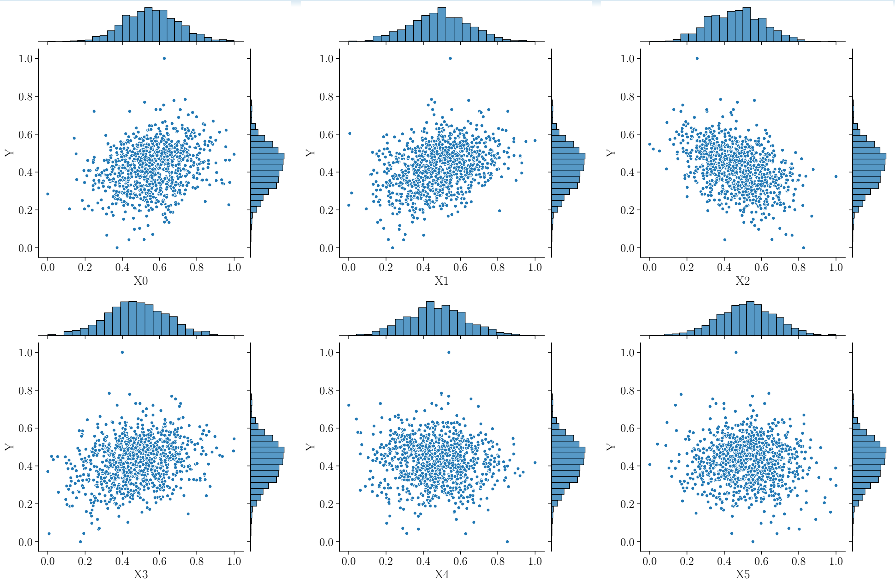


Figure: Test-case CATHARE: inputs output scatter plots, part 1 ($N = 10^3$)

Industrial CATHARE use-case

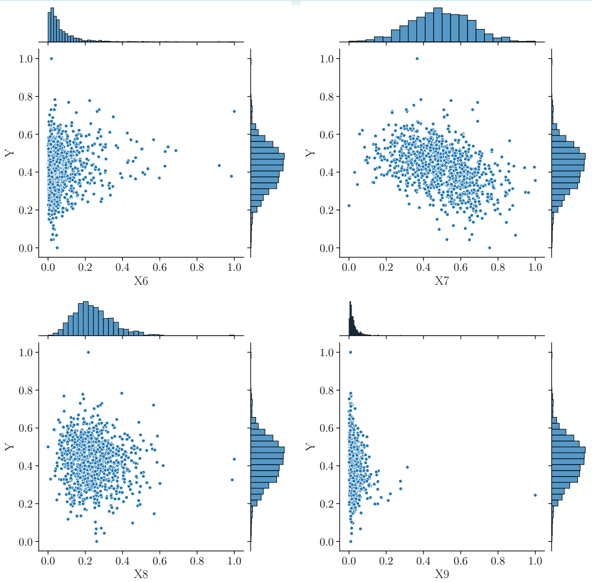


Figure: Test-case CATHARE: inputs output scatter plots, part 2 ($N = 10^3$)