# Options for high-dimensional Bayesian optimization

Mickaël Binois (Acumes team - Inria Center at Université Côte d'Azur)
mickael.binois@inria.fr

joint work with V. Picheny (Secondmind), N. Wycoff (Georgetown University)

ANR SAMOURAI, Paris

December 11th, 2024

# Problem description

Let us consider an expensive-to-evaluate **black box** simulator:

$$f : \mathbf{X} \subset \mathbb{R}^d \to \mathbb{R}.$$

Suppose we want to minimize $f$: find $\mathbf{x}^* \in \operatorname*{argmin}_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$.

Here, $\mathbf{X} = [-1, 1]^d$, corresponding to box constraints.

In addition $d$ is possibly *large*[1], especially with respect to the evaluation budget.

Common occurrence in Physics, Operations Research, Epidemiology, Machine Learning, ...

[1] What large means is very much application dependent. In general, $d > 10$ is considered as large in BO.

# Outline

1. Background on Bayesian optimization

2. High-dimensional GPs
   - Overview
   - Additive models
   - Active subspace estimation
   - Other methods

3. Reconciling linear embedding and additive models?

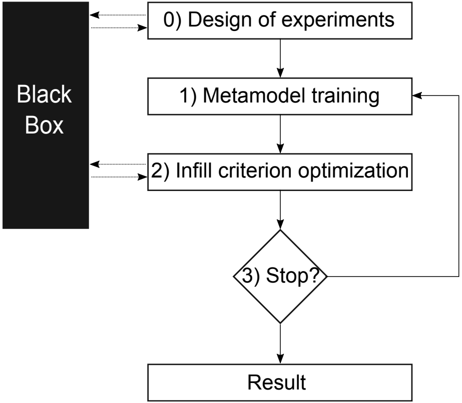
4. Application to optimization

5. Conclusion

# General solving procedure in Bayesian Optimization (BO)[2],[3]

**Bayesian optimization**

Sequential design strategy based on a distribution over functions to define an acquisition function.



For instance:

0. Maximin Latin Hypercubes Samples
1. Gaussian process model
2. Expected Improvement
3. Budget

[2] J. Mockus. *Bayesian approach to global optimization*. Springer, 1989.

[3] R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2022.

# Gaussian process regression

We use a zero mean GP prior on $y$, with covariance $k$: $Y \sim \mathcal{GP}(0, k)$.

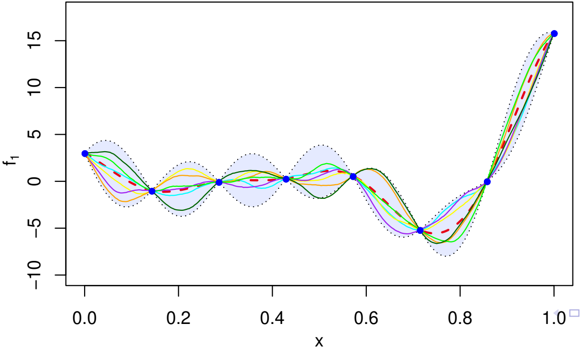
MVN conditional identities give directly the result on $(\mathbf{x}_i, y_i)_{1 \leq i \leq n}$:

$$Y|\mathbf{y} \sim \mathcal{GP}(\mu, \sigma^2) \text{ with}$$

$$m_n(\mathbf{x}) = \mathbb{E}(Y(\mathbf{x})|\mathbf{y}) = \mathbf{k}(\mathbf{x})^\top \mathbf{K}_N^{-1} \mathbf{y},$$

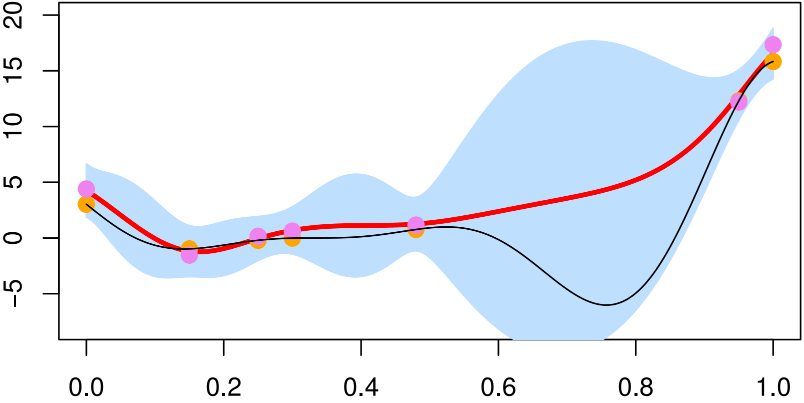
$$s_n^2(\mathbf{x}) = \mathbb{V}\mathrm{ar}(Y(\mathbf{x})|\mathbf{y}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^\top \mathbf{K}_N^{-1} \mathbf{k}(\mathbf{x}), \text{ where}$$

$\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n))^\top$, $\mathbf{K}_N = (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i,j \leq n}$.



# Noisy observations

GPs readily handle Gaussian noise, e.g., through the estimation of a constant noise term.



For the reminder of the talk, we do not dvelve more on this additional challenge.

# GP training

GPs have their own hyperparameters, mostly for the kernel function.
The most popular kernels are stationary, e.g., the Gaussian kernel:
$k(x, x'|\tau^2, \theta) = \tau^2 \exp(-(x-x')^2/\theta) = \tau^2 c(\mathrm{abs}(x-x')|\tau^2, \theta)$.

Hyperparameter estimation can be based on:

- model error (i.e., cross validation, training/testing sets)
- variogram analysis
- (log)-likelihood, possibly regularized (maximum a posteriori)

Likelihood, i.e., multivariate normal density:

$$L = \frac{1}{(2\pi)^{n/2}|\mathbf{K}_n|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{y}^\top \mathbf{K}_n^{-1}\mathbf{y}\right).$$

Alternatives include maximum-likelihood estimation and more Bayesian versions with various degrees of approximation.

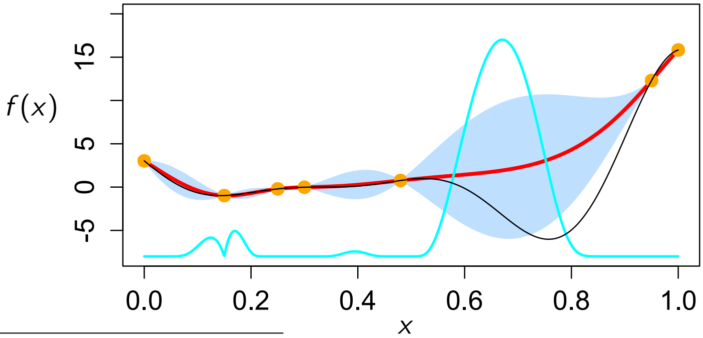
# Infill criterion - Expected Improvement[4]

Improvement: $I : \mathbf{x} \in \mathbf{X} \to \max\{f^* - Y(\mathbf{x}), 0\} \in \mathbb{R},\ f^* = \min_{1 \le i \le n} f(\mathbf{x}_i)$

## Expected Improvement

$$E[I(\mathbf{x})|\mathbf{y}] = (f^* - m_n(\mathbf{x}))\,\Phi\left(\frac{f^* - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right) + s_n(\mathbf{x})\phi\left(\frac{f^* - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right)$$

→ balance between exploration and exploitation



[4] J. Mockus, V. Tiesis, and A. Zilinskas. "The application of Bayesian methods for seeking the extremum". In: *Towards Global Optimization* 2.117-129 (1978), p. 2.

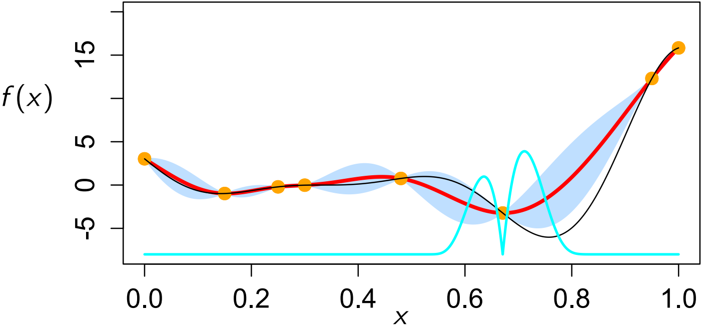
# Infill criterion - Expected Improvement[4]

Improvement: $I : \mathbf{x} \in \mathbf{X} \to \max\{f^* - Y(\mathbf{x}), 0\} \in \mathbb{R},\ f^* = \min_{1\leq i\leq n} f(\mathbf{x}_i)$

**Expected Improvement**

$$E[I(\mathbf{x})|\mathbf{y}] = (f^* - m_n(\mathbf{x}))\,\Phi\left(\frac{f^* - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right) + s_n(\mathbf{x})\phi\left(\frac{f^* - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right)$$

→ balance between exploration and exploitation



[4] Mockus, Tiesis, and Zilinskas, "The application of Bayesian methods for seeking the extremum".

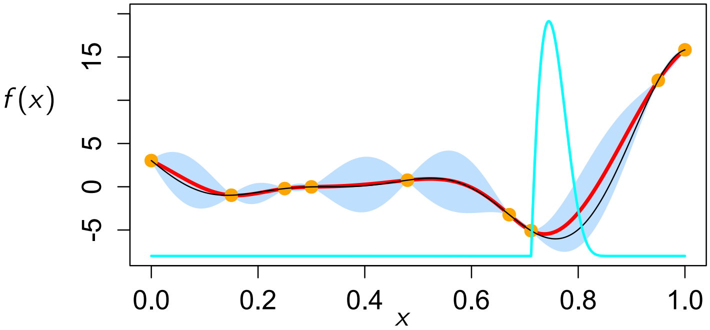
# Infill criterion - Expected Improvement[4]

Improvement: $I : \mathbf{x} \in \mathbf{X} \rightarrow \max\{f^* - Y(\mathbf{x}), 0\} \in \mathbb{R}$, $f^* = \min_{1 \leq i \leq n} f(\mathbf{x}_i)$

## Expected Improvement

$$E[I(\mathbf{x})|\mathbf{y}] = (f^* - m_n(\mathbf{x}))\,\Phi\left(\frac{f^* - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right) + s_n(\mathbf{x})\phi\left(\frac{f^* - m_n(\mathbf{x})}{s_n(\mathbf{x})}\right)$$

→ balance between exploration and exploitation



[4] Mockus, Tiesis, and Zilinskas, "The application of Bayesian methods for seeking the extremum".

# Outline

1 Background on Bayesian optimization

2 High-dimensional GPs
- Overview
- Additive models
- Active subspace estimation
- Other methods

3 Reconciling linear embedding and additive models?

4 Application to optimization

5 Conclusion

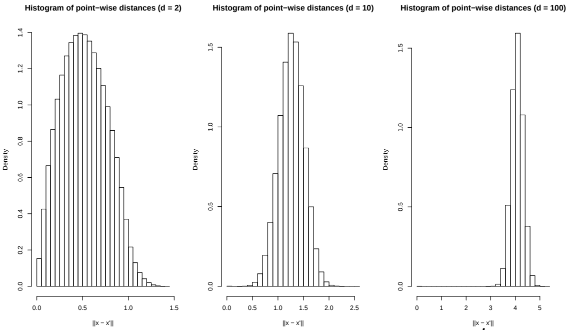
# Effects of the curse of dimensionality

A regular grid of 10 points in dimension $d$ requires $10^d$ points.

Most of the volume concentrates on the boundary of the domain:

- volume(unit $d$-sphere)/volume(unit $d$-cube) $\to 0$ as $d \to \infty$
- volume($d$-ball of radius $(1-\delta)R$)/volume($d$-ball of radius $R$) $= o((1-\delta)^d)$

Uniformly sampled points are far away from each other:



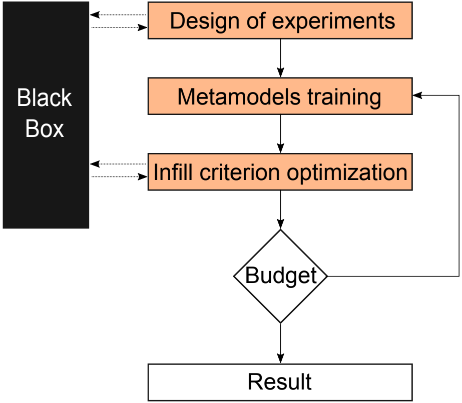
Issue for most kernels, e.g. $k(\mathbf{x},\mathbf{x}') = c(\|\mathbf{x}-\mathbf{x}'\|)$, $k(\mathbf{x},\mathbf{x}') = \prod_{i=1}^{d} k_i(x_i, x_i')$

$(0.9^{10} \approx 0.35, 0.95^{30} \approx 0.21)$

# … and consequences

These high dimensional effects impact all steps of BO:

0. distances for maximin LHS (unless projected[a])
1. distances in the GP model (+ training)
2. optimizing Expected Improvement



[a] V. R. Joseph, E. Gul, and S. Ba. “Maximum projection designs for computer experiments”. In: *Biometrika* 102.2 (2015), pp. 371–380.

# What to do?

The main option is to assume additional structural information:

- some variables have no influence (screening);
- the problem is intrinsically of lower dimension (linear/non-linear embeddings);
- or via additivity and functional ANOVA decompositions.

More exotic structures are also possible.

A review is available in[5].

[5] M. Binois and N. Wycoff. "A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization". In: *ACM Transactions on Evolutionary Learning and Optimization* 2.2 (2022), pp. 1–26.

# Standard GP Training adaptations

Defaults for GP packages are often thought for low numbers of variables, and with the flat limit[6]. Recent discussions include:

- Appropriate priors for the MAP or MLE (in particular the upper bounds)[7],[8],[9].
- Use of Matérn kernel (not squared distance), diffuse priors, UCB[10].
- Robust multi-objective fit beyond MLE (LOO, coverage, . . . )[11],[12].
- Not trying to learn the hyperparameters[13]

These may also hint that more complex structure is even harder to learn.

[6] S. Barthelmé et al. "Gaussian process regression in the flat limit". In: *The Annals of Statistics* 51.6 (2023), pp. 2471–2505.

[7] D. Eriksson and M. Jankowiak. "High-dimensional Bayesian optimization with sparse axis-aligned subspaces". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 493–503.

[8] C. Hvarfner, E. O. Hellsten, and L. Nardi. "Vanilla Bayesian Optimization Performs Great in High Dimension". In: *arXiv preprint arXiv:2402.02229* (2024).

[9] M. Gu, X. Wang, and J. O. Berger. "Robust Gaussian stochastic process emulation". In: *The Annals of Statistics* 46.6A (2018), pp. 3038–3066.

[10] Z. Xu and S. Zhe. "Standard Gaussian Process is All You Need for High-Dimensional Bayesian Optimization". In: *arXiv preprint arXiv:2402.02746* (2024).

[11] A. Marrel and B. Iooss. "Probabilistic surrogate modeling by Gaussian process: A review on recent insights in estimation and validation". In: *Reliability Engineering & System Safety* (2024), p. 110094.

[12] A. Marrel and B. Iooss. "Probabilistic surrogate modeling by Gaussian process: A new estimation algorithm for more robust prediction". In: *Reliability Engineering & System Safety* 247 (2024), p. 110120.

[13] T. Appriou, D. Rullière, and D. Gaudrie. "Combination of optimization-free kriging models for high-dimensional problems". In: *Computational Statistics* (2023), pp. 1–23.

# Scaling up to many variables: inactive variables / single index

One simple attempt to tackle high-dimension is to assume that most of the variables have no effect (or are handled as noise):

$$\text{model: } f(\mathbf{x}) = g(\mathbf{x}_I) \text{ with } I \subset \{1, \ldots, d\}, |I| \ll d$$

and then identify them sequentially, (see e.g.,[14],[15],[16],[17]).

Another popular dimension reduction technique is the single index model:

$$f(\mathbf{x}) = g(\mathbf{a}^\top \mathbf{x}) \text{ with } \mathbf{a} \in \mathbb{R}^d$$

See e.g.,[18] for the GP treatment.

[14] A. Marrel et al. "An efficient methodology for modeling complex computer codes with Gaussian processes". In: *Computational Statistics & Data Analysis* 52.10 (2008), pp. 4731–4744.

[15] B. Chen, R. Castro, and A. Krause. "Joint optimization and variable selection of high-dimensional Gaussian processes". In: *Proc. International Conference on Machine Learning (ICML)*. 2012.

[16] M. B. Salem et al. "Sequential dimension reduction for learning features of expensive black-box functions". In: (2018).

[17] A. Spagnol, R. L. Riche, and S. D. Veiga. "Global sensitivity analysis for optimization with variable selection". In: *SIAM/ASA Journal on uncertainty quantification* 7.2 (2019), pp. 417–443.

[18] R. B. Gramacy and H. Lian. "Gaussian process single-index models as emulators for computer experiments". In: *Technometrics* 54.1 (2012), pp. 30–41.

# Scaling-up to many variables: additive models

Model: $f(\mathbf{x}) = \sum_{i=1}^{d} g_i(x_i)$, see, e.g.,[19],[20].

For GPs, amounts to summing univariate kernels: $k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{d} k_i(x_i, x_i')$

Pros:

- scale linearly with $d$
- predictive mean is the sum of univariate predictive means
- optimization of the acquisition function is simplified
- interpretability

Cons:

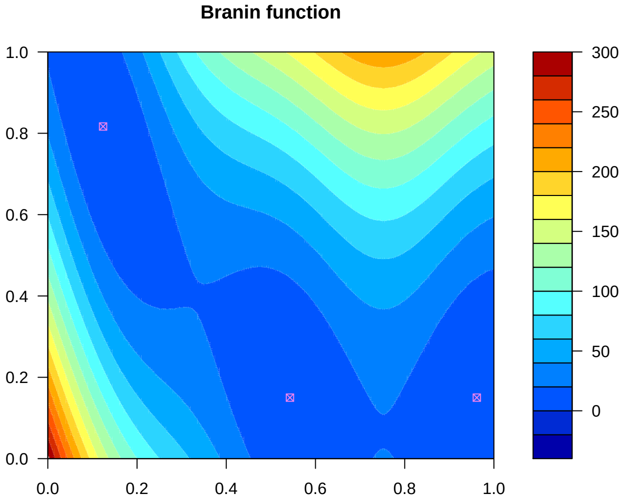
- zero predictive variance at unobserved points
- training is harder ($2 \times d + 1$ hyperparameters)
- very strong structural assumption

[19] N. Durrande, D. Ginsbourger, and O. Roustant. “Additive Kernels for Gaussian Process Modeling”. In: *Annales de la Facultée de Sciences de Toulouse* (2012), p. 17.

[20] D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen. “Additive Gaussian processes”. In: *NeurIPS*. 2011, pp. 226–234.

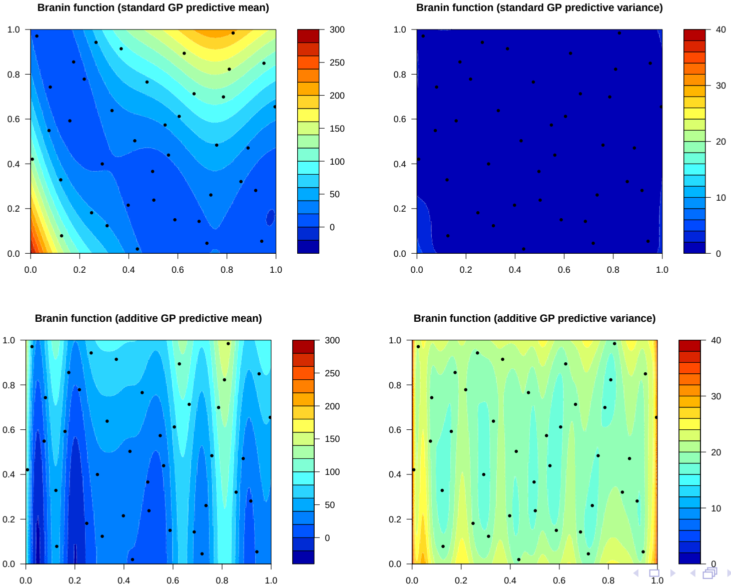
# Scaling-up to many variables: additive models

Example: 2D Branin function with 40 design points



# Scaling-up to many variables: additive models

## Example: 2D Branin function with 40 design points



Branin function (standard GP predictive mean)

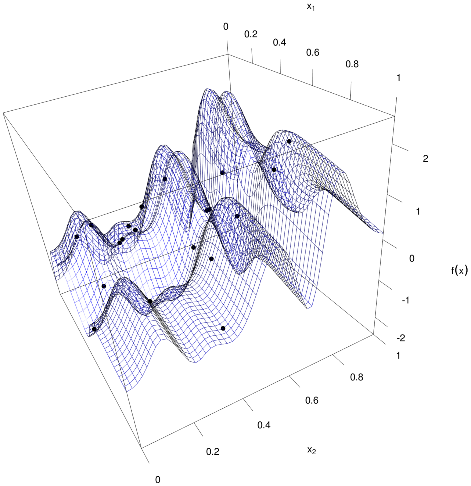
Branin function (standard GP predictive variance)

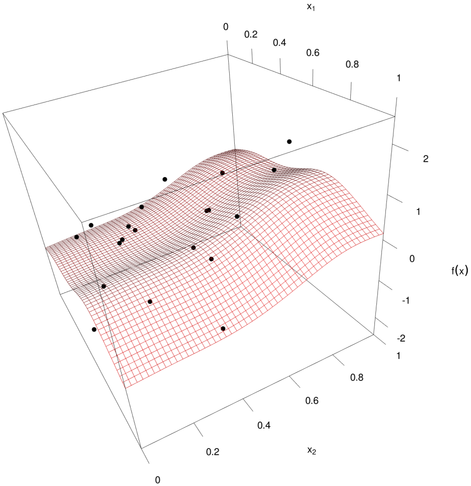
Branin function (additive GP predictive mean)

Branin function (additive GP predictive variance)

# Scaling-up to many variables: additive models

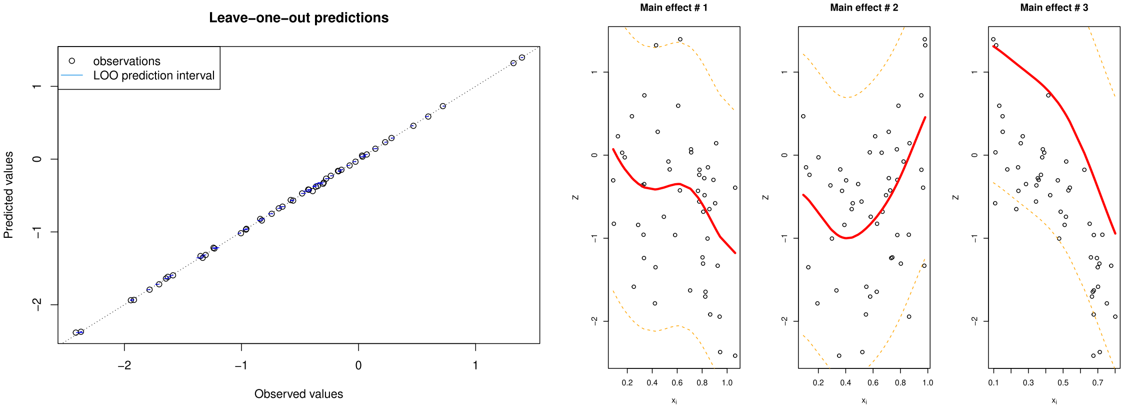
## Example: First order additive is still flexible





# Scaling-up to many variables: additive models

## Example: Additive function with main effects



# Scaling up to many variables: additive by-part models

Additivity can be extended to groups of variables, see e.g.,[19],[20]: $f(\mathbf{x}) = \sum_{i=1}^{M} g^{(i)}(\mathbf{x}^{(i)})$ with $A_i$ disjoint subsets of $\{1, \dots, d\}$.

The non-overlapping case is addressed, e.g., by[21], but inference is difficult (especially with BO).

Subsequent works try learning a tree decomposition of the variables[22], that can be local, random and data independent[23].

The underlying dependence graphs between variables are more or less rich depending on the assumptions.

[19] K. Kandasamy, J. Schneider, and B. Póczos. "High dimensional Bayesian optimisation and bandits via additive models". In: (2015), pp. 295–304.

[20] Z. Wang et al. "Batched Large-scale Bayesian Optimization in High-dimensional Spaces". In: *AISTATS*. 2018.

[21] P. Rolland et al. "High-Dimensional Bayesian Optimization via Additive Models with Overlapping Groups". In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Vol. 84. Proceedings of Machine Learning Research. PMLR, 2018, pp. 298–307.

[22] E. Han, I. Arora, and J. Scarlett. "High-dimensional Bayesian optimization via tree-structured additive models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 9. 2021, pp. 7630–7638.

[23] J. K. Ziomek and H. B. Ammar. "Are random decompositions all we need in high dimensional Bayesian optimisation?" In: *International Conference on Machine Learning*. PMLR. 2023, pp. 43347–43368.

# Scaling up to many variables: additive by-part models (2)

Higher order structure illustrations:

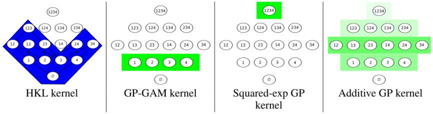
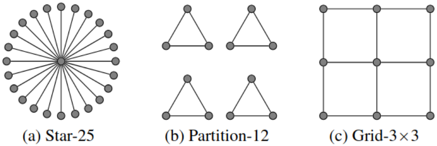


Figure 3: A comparison of different models. Nodes represent different interaction terms, ranging from first-order to fourth-order interactions. Far left: HKL can select a hull of interaction terms, but must use a pre-determined weighting over those terms. Far right: the additive GP model can weight each order of interaction seperately. Neither the HKL nor the additive model dominate one another in terms of flexibility, however the GP-GAM and the SE-GP are special cases of additive GPs.

from[24];



(a) Star-25 (b) Partition-12 (c) Grid-3×3

Figure 4: Synthetic Dependency Graphs Structures.

from[25]

A useful tool: Newton-Girard formula[26]: $\mathcal{O}(d^2)$ computation of the full interaction kernel, rather than $\mathcal{O}(2^d)$ (given that high order kernels are product of low order ones: $k((x_1, x_2), (x_1', x_2')) = k_1(x_1, x_1')k_2(x_2, x_2')$).

[24] Duvenaud, Nickisch, and Rasmussen, “Additive Gaussian processes”.

[25] Han, Arora, and Scarlett, “High-dimensional Bayesian optimization via tree-structured additive models”.

[26] Duvenaud, Nickisch, and Rasmussen, “Additive Gaussian processes”.

# Scaling up to many variables: ANOVA

An alternative formulation is the Functional ANOVA decomposition[27]:

$$f(\mathbf{x}) = c + \sum_{i=1}^{d} g_i(x_i) + \sum_{j<k} g_{jk}(x_j, x_k) + \cdots + f_{12\ldots d}(x_1, x_2, \ldots, x_d) \text{ with orthogonal terms } g_{\ldots}$$

Kernel $k_{ANOVA}(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{d} (1 + k^i(x_i, x_i'))$[28,29], also revisited by[30]

More flexible framework but estimation is harder (up to $2^d - 1$ terms).

Nice link with sensitivity analysis.

Going further: separating the additive and non-additive parts[31,32]

[27] T. Muehlenstaedt et al. "Data-driven Kriging models based on FANOVA-decomposition". In: *Statistics and Computing* 22.3 (2012), pp. 723–738.

[28] M. Stitson et al. "Support vector regression with ANOVA decomposition kernels". In: *Advances in kernel methods—Support vector learning* (1999), pp. 285–292.

[29] D. Ginsbourger et al. "On ANOVA decompositions of kernels and Gaussian random field paths". In: *Monte Carlo and Quasi-Monte Carlo Methods*. Springer, 2016, pp. 315–330.

[30] X. Lu, A. Boukouvalas, and J. Hensman. "Additive Gaussian Processes Revisited". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 14358–14383.

[31] N. Lenz. *Additivity and Ortho-Additivity in Gaussian Random Fields*. Tech. rep. Aug. 2013. URL: https://hal.science/hal-01063741.

[32] Ginsbourger et al., "On ANOVA decompositions of kernels and Gaussian random field paths".

# A promising framework

Lu et al. (2024)[33] revisits the key ingredients to identify high-order interactions:

- the Newton-Girard formulation (see e.g.,[34]) of high order terms, for speed and inference (with limitations)
- one variance term per interaction level
- orthogonality conditions: $\int_{D_I} f_I(\mathbf{x}_I)p_I(\mathbf{x}_I)d\mathbf{x}_I = 0$ to correct identifiability issues.

It comes with a `Python` implementation. In `R`, there are the `fanovaGraph`[35] and `kergp`[36] packages.

[33] Lu, Boukouvalas, and Hensman, "Additive Gaussian Processes Revisited".

[34] Duvenaud, Nickisch, and Rasmussen, "Additive Gaussian processes".

[35] J. Fruth et al. *fanovaGraph: Building Kriging Models from FANOVA Graphs*. R package version 1.5. 2020. URL: https://CRAN.R-project.org/package=fanovaGraph.

[36] Y. Deville, D. Ginsbourger, and O. R. C. N. Durrande. *kergp: Gaussian Process Laboratory*. R package version 0.5.7. 2024. URL: https://CRAN.R-project.org/package=kergp.

# Active learning?

Not much work is dedicated to learning the additive structure sequentially.

For the first order linear model, optimal DoEs are product of univariate ones[37].

For higher order models, perhaps using the vanishing variance property at unvisited designs is useful? E.g., similar to the split and doubt strategy from[38].

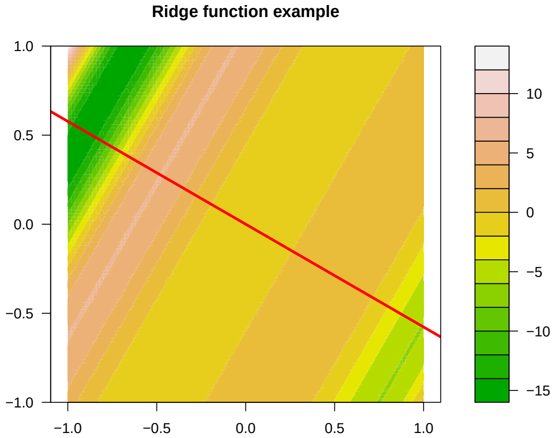
[37] R. Schwabe. "Designing experiments for additive nonlinear models". In: *MODA4—Advances in Model-Oriented Data Analysis: Proceedings of the 4th International Workshop in Spetses, Greece June 5–9, 1995*. Springer. 1995, pp. 77–85.

[38] Salem et al., "Sequential dimension reduction for learning features of expensive black-box functions".

# Scaling up to many variables: active subspaces

**Observation**: the variation is often concentrated around a few unknown directions $r \ll d$

Model: $f(\mathbf{x}) = g(\mathbf{A}^\top \mathbf{x})$ with $\mathbf{A} \in \mathbb{R}^{d \times r}$ (ridge function)



# Scaling up to many variables: active subspaces

**Observation**: the variation is often concentrated around a few unknown directions $r \ll d$

$$\text{Model: } f(\mathbf{x}) = g(\mathbf{A}^\top \mathbf{x}) \text{ with } \mathbf{A} \in \mathbb{R}^{d \times r} \text{ (ridge function)}$$

Backed by empirical and theoretical evidence, e.g.,[39]
Options exist to estimate **A**, most rely either:

- on the gradient of $f$, to estimate $\mathbf{C} = \int \nabla(f(\mathbf{x}))^\top \nabla(f(\mathbf{x}))\mu(dx)$, see e.g.,[40],[41].
- on treating **A** as an hyperparameter, see e.g.,[42],[43],[44];
- on using PCA[45] or PLS[46].

[39] P. G. Constantine, Z. del Rosario, and G. Iaccarino. "Many physical laws are ridge functions". In: *arXiv:1605.07974* (2016).

[40] J. Djolonga, A. Krause, and V. Cevher. "High-Dimensional Gaussian Process Bandits". In: *NIPS*. 2013, pp. 1025–1033.

[41] P. G. Constantine. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM, 2015.

[42] R. Garnett, M. A. Osborne, and P. Hennig. "Active learning of linear embeddings for Gaussian processes". In: *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. AUAI Press. 2014, pp. 230–239.

[43] R. Tripathy, I. Bilionis, and M. Gonzalez. "Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation". In: *Journal of Computational Physics* 321 (2016), pp. 191 –223.

[44] P. Marcy. "Bayesian Gaussian Process Models for Dimension Reduction Uncertainties". ASA Joint research conference. 2018.

[45] E. Raponi et al. "High dimensional Bayesian optimization assisted by principal component analysis". In: *PPSN*. Springer. 2020, pp. 169–183.

[46] M. A. Bouhlel et al. "Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction". In: *Structural and Multidisciplinary Optimization* 53.5 (2016), pp. 935–952.

# Active subspace methodology[48]

The key quantity for active subspaces is $\mathbf{C} = \int_d \nabla(f(\mathbf{x}))\nabla(f(\mathbf{x}))^\top \mu(d\mathbf{x})$ where $\mu$ is a user defined measure.

### Active subspace framework

```
Require: $d$, $M$, $r$ (optional)
    Draw $M$ iid samples $\mathbf{x}_i \sim \mu$.
    Compute $\nabla f(\mathbf{x}_i)$.
    Compute $\widehat{\mathbf{C}} = \frac{1}{M}\sum_{i=1}^{M}(\nabla f(\mathbf{x}_i))(\nabla f(\mathbf{x}_i))^\top$ and its eigen value decomposition $\widehat{\mathbf{C}} = \widehat{\mathbf{W}} \cdot \mathrm{Diag}(\lambda_1, \dots, \lambda_d) \cdot \widehat{\mathbf{W}}^\top$.
    (Optional) Define $r$ based on eigen-value gaps.
    Perform the task in the reduced rotated basis $\mathbf{W}_{1,\dots,r}$.
```

When $f$ is an expensive black-box, this can be done on a surrogate, with two bonuses:

- it works on black-boxes with no derivatives[47];
- it alleviates the iid restriction.

[47] P. S. Palar and K. Shimoyama. "On The Accuracy of Kriging Model in Active Subspaces". In: *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference.* 2018, p. 0913.

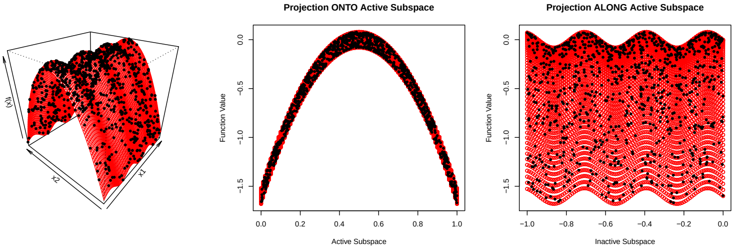
[48] Constantine, *Active subspaces: Emerging ideas for dimension reduction in parameter studies.*

# Illustration example 1

Consider the function $f(x_1, x_2) = a\sin(bx_1) + cx_2^2$ with $a = 0.1, b = 20, c = -4$ on the unit square.

Large eigen-value gap:

$$\hat{\mathbf{C}} = \begin{bmatrix} -0.00 & -0.99 \\ 0.99 & -0.00 \end{bmatrix} \begin{bmatrix} 13.63 & 0 \\ 0 & 1.30 \end{bmatrix} \begin{bmatrix} -0.00 & 0.99 \\ -0.99 & -0.00 \end{bmatrix}$$



Different from the Automatic Relevance Determination principle (see, e.g.,[49]) of keeping variables with the smallest estimated lengthscales.

[49] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. URL: http://www.gaussianprocess.org/gpml/.

# Closed form expression for $\mathbf{C}$[50]

Assuming that k is twice differentiable, the joint distribution of $(Y(\mathbf{X}), \partial Y(\mathbf{x})/\partial \mathbf{x}_1, \ldots, \partial Y(\mathbf{x})/\partial \mathbf{x}_d)$ is:

$$\begin{pmatrix} \mathbf{y}_n \\ \partial Y(\mathbf{x})/\partial \mathrm{x}_1 \\ \vdots \\ \partial Y(\mathbf{x})/\partial \mathrm{x}_d \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} \mathbf{0_n} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{K}_n & \partial \mathbf{k}(\mathbf{x})^\top/\partial \mathrm{x}_1 & \ldots & \partial \mathbf{k}(\mathbf{x})^\top/\partial \mathrm{x}_d \\ \partial \mathbf{k}(\mathbf{x})/\partial \mathrm{x}_1 & \partial^2 k(\mathbf{x},\mathbf{x})/\partial \mathrm{x}_1^2 & \ldots & \partial^2 k(\mathbf{x},\mathbf{x})/\partial \mathrm{x}_1 \partial \mathrm{x}_d \\ \vdots & \vdots & \ddots & \vdots \\ \partial \mathbf{k}(\mathbf{x}) \partial / \mathrm{x}_d & \partial^2 k(\mathbf{x},\mathbf{x})/\partial \mathrm{x}_d \partial \mathrm{x}_1 & \ldots & \partial^2 k(\mathbf{x},\mathbf{x})/\partial \mathrm{x}_d^2 \end{pmatrix} \right)$$

$$\text{In shorthand: } \begin{pmatrix} \mathbf{y}_n \\ \nabla Y(\mathbf{x}) \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} \mathbf{0_n} \\ \mathbf{0_d} \end{pmatrix}, \begin{pmatrix} \mathbf{K}_n & \kappa(\mathbf{x})^\top \\ \kappa(\mathbf{x}) & \mathbf{K}_d(\mathbf{x}) \end{pmatrix} \right)$$

As a result, $\nabla Y(\mathbf{x}) | \mathcal{A}_n \sim \mathcal{N}(\mu_n(\mathbf{x}), \kappa_n(\mathbf{x}, \mathbf{x}))$ with:

$$\mu_n(\mathbf{x}) = \kappa(\mathbf{x}) \mathbf{K}_n^{-1} \mathbf{y}_n$$

$$\kappa_n(\mathbf{x}, \mathbf{x}') = \mathbf{K}_d(\mathbf{x}, \mathbf{x}') - \kappa(\mathbf{x}) \mathbf{K}_n^{-1} \kappa(\mathbf{x}')^\top$$

$C_{ij}^{(n)} = E_{ij} - tr\left(\mathbf{K}_n^{-1} \mathbf{W}_{ij}\right) + \mathbf{y}_n^\top \mathbf{K}_n^{-1} \mathbf{W}_{ij} \mathbf{K}_n^{-1} \mathbf{y}_n$ where $\mathbf{W}_{ij} = \int_{\mathcal{X}} \kappa_i(X) \kappa_j(X)^\top d\mu$ and $E_{ij} = \int_{\mathcal{X}} \frac{\partial^2 k(X,X)}{\partial x_i \partial x_j} d\mu$.

[50] N. Wycoff, M. Binois, and S. M. Wild. "Sequential Learning of Active Subspaces". In: *Journal of Computational and Graphical Statistics* 30.4 (2021), pp. 1224–1237.

# Closed form expression for $\mathbf{C}$ (cont'd)

**Closed-form $\mathbf{C}$ expression for a GP**

$C_{ij}^{(n)} = E_{ij} - tr\left(\mathbf{K}_n^{-1}\mathbf{W}_{ij}\right) + \mathbf{y}_n^\top \mathbf{K}_n^{-1}\mathbf{W}_{ij}\mathbf{K}_n^{-1}\mathbf{y}_n$ where $\mathbf{W}_{ij} = \int_{\mathcal{X}} \kappa_i(X)\kappa_j(X)^\top d\mu$ and $E_{ij} = \int_{\mathcal{X}} \frac{\partial^2 k(X,X)}{\partial x_i \partial x_j} d\mu$.

Balance between Integrated Mean Squared Prediction Error (IMSPE) and covariance between partial derivatives means.

The diagonal terms correspond to derivative-based global sensitivity measures, see, e.g.,[51].

Given $n$ observations, $\mathbf{C}^{(n)}$ only depends on the kernel hyperparameters.

This allows a one-shot learning procedure of a GP with dimension reduction[52].

[51] M. De Lozzo and A. Marrel. "Estimation of the derivative-based global sensitivity measures using a Gaussian process metamodel". In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1 (2016), pp. 708–738.

[52] M. Binois and V. Picheny. "Combining additivity and active subspaces for high-dimensional Gaussian process modeling". In: *arXiv preprint arXiv:2402.03809* (2024).

# Sequential version

## Updating $\mathbf{C}^n$

Given a new design point $\tilde{\mathbf{x}}$ but not the function value at this location, i.e., $y_{n+1} \sim \mathcal{N}(m_n(\tilde{\mathbf{x}}), k_n(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}))$, the random variable $C_{ij}^{(n+1)} - C_{ij}^{(n)}$, can be written as:

$$\begin{aligned} &= -\left(\mathbf{w}_a(\tilde{\mathbf{x}}) + \mathbf{w}_b(\tilde{\mathbf{x}})\right)^\top \mathbf{g}(\tilde{\mathbf{x}}) - \sigma_n^2(\tilde{\mathbf{x}})^{-1} \left[w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) + \mathbf{k}_n(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1} \mathbf{W}_n \mathbf{K}_n^{-1} \mathbf{k}_n(\tilde{\mathbf{x}})\right] \\ &\quad + Z\sigma_n(\tilde{\mathbf{x}})^{-1} \left[\mathbf{y}_n^\top \mathbf{K}_n^{-1} \mathbf{W}_n \mathbf{K}_n^{-1} \mathbf{k}_n(\tilde{\mathbf{x}}) + \mathbf{k}_n(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1} \mathbf{W}_n \mathbf{K}_n^{-1} \mathbf{y}_n - \left(\mathbf{w}_a(\tilde{\mathbf{x}}) + \mathbf{w}_b(\tilde{\mathbf{x}})\right)^\top \mathbf{K}_n^{-1} \mathbf{y}_n\right] \\ &\quad + Z^2 \sigma_n^2(\tilde{\mathbf{x}})^{-1} \left[w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) + \mathbf{k}_n(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1} \mathbf{W}_n \mathbf{K}_n^{-1} \mathbf{k}_n(\tilde{\mathbf{x}}) - \left(\mathbf{w}_a(\tilde{\mathbf{x}}) + \mathbf{w}_b(\tilde{\mathbf{x}})\right)^\top \mathbf{K}_n^{-1} \mathbf{k}_n(\tilde{\mathbf{x}})\right] \\ &\quad := \alpha_{i,j}(\tilde{\mathbf{x}}) + Z\beta_{i,j}(\tilde{\mathbf{x}}) + Z^2 \beta_{i,j}(\tilde{\mathbf{x}}) \end{aligned}$$

with $Z \sim \mathcal{N}(0,1)$, $\mathbf{g}(\tilde{\mathbf{x}}) = -\sigma_n^2(\tilde{\mathbf{x}})^{-1} \mathbf{K}_n^{-1} \mathbf{k}_n(\tilde{\mathbf{x}})$, $w_a(\tilde{\mathbf{x}}) = W_{ij}(\tilde{\mathbf{x}}, \mathbf{X})$, and $w_b(\tilde{\mathbf{x}}) = W_{ji}(\tilde{\mathbf{x}}, \mathbf{X})$

It remains to define an appropriate criterion for sequential design: an uncertainty measure $J(\tilde{\mathbf{x}})$ on (e.g., Expected Improvement, entropy, $\cdots$ for optimization)

# Sequential design criteria definitions

A natural way to proceed is to consider a variance of $\mathbf{C}^{(n+1)}$.

**Criteria $J(\tilde{\mathbf{x}})$**

$$\begin{aligned}
\texttt{Trace} &= \mathbb{V}ar\left(\mathrm{tr}(\mathbf{C}^{(n+1)})\right) \\
\texttt{Var1} &= ||\mathbb{E}[(\mathbf{C}^{(n+1)} - \mathbb{E}[\mathbf{C}^{(n+1)}]) \odot (\mathbf{C}^{(n+1)} - \mathbb{E}[\mathbf{C}^{(n+1)}])]||_F^2 \\
\texttt{Var2} &= ||\mathbb{E}[(\mathbf{C}^{(n+1)} - \mathbb{E}[\mathbf{C}^{(n+1)}])(\mathbf{C}^{(n+1)} - \mathbb{E}[\mathbf{C}^{(n+1)}])]||_F^2
\end{aligned}$$

Closed form expressions and derivatives are available.

Finding $\arg_{\tilde{\mathbf{x}}\in D}\min J(\tilde{\mathbf{x}})$ is irrelevant.

Inverted Step-wise Uncertainty Reduction strategy, see e.g.,[53]:
find the design $\tilde{\mathbf{x}}$ that perturbs $\mathbf{C}^{(n+1)}$ the most:

$$\tilde{\mathbf{x}}^* \in \arg_{\tilde{\mathbf{x}}\in D}\max J(\tilde{\mathbf{x}})$$

[53] T. Labopin-Richard and V. Picheny. "Sequential design of experiments for estimating quantiles of black-box functions". In: *Statistica Sinica* (2018), pp. 853–877.

# Sequential design summary

### Pseudo code for the sequential active subspace learning

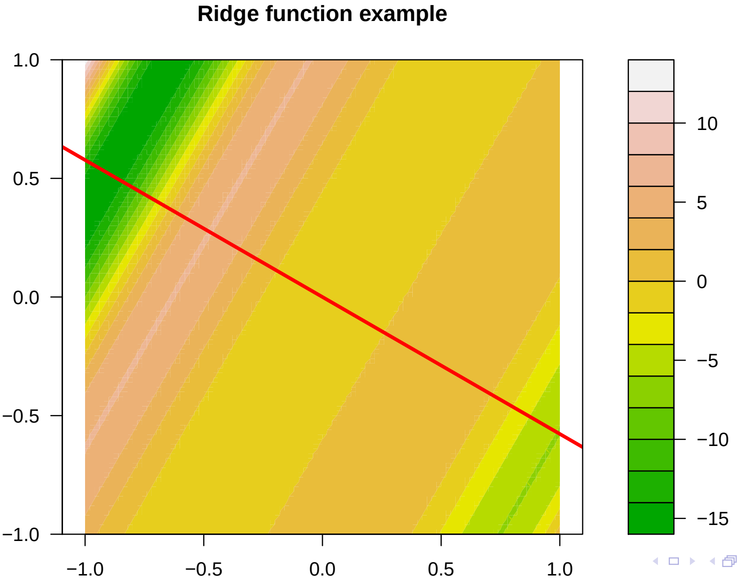
```
Require: $n_0$.
Construct an initial design of experiments in $\mathcal{X}$, of size $n_0$.
Build the GP model with kernel $k$.
while time/evaluation budget not exhausted do
    Find $\tilde{\mathbf{x}}^* \in \arg\max_{\mathbf{x} \in D} J(\mathbf{x})$
    Evaluate the objective function at $\mathbf{y}_{n+1}$, $f(\tilde{\mathbf{x}})$.
    Update the GP model based on new data.
end while
```

The estimation of AS from a GP is available in the `activegp`[54] `R` package.

[54] N. Wycoff and M. Binois. *activegp: Gaussian Process Based Design and Analysis for the Active Subspace Method*. R package version 1.1.1. 2024. URL: https://CRAN.R-project.org/package=activegp.

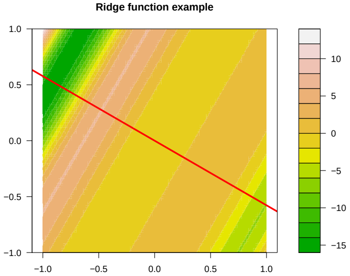
# Illustrative 2d example

10 initial design points + 20 sequential points

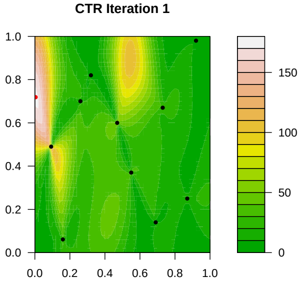


# Illustrative 2d example

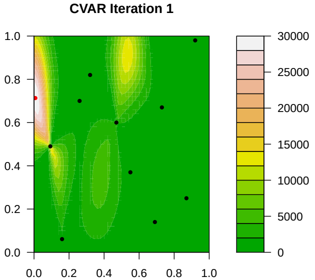
10 initial design points + 20 sequential points



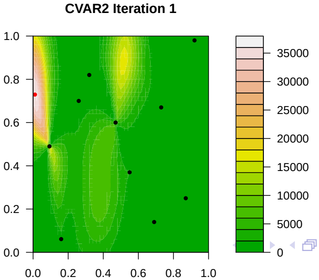
Ridge function example



CTR Iteration 1

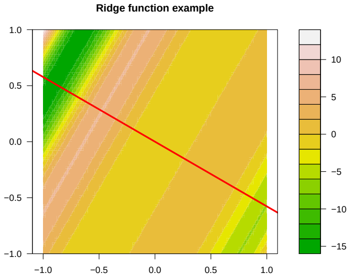


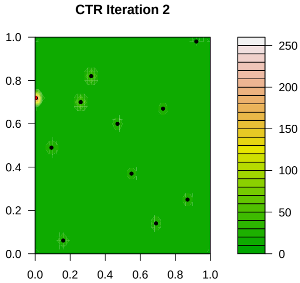
CVAR Iteration 1

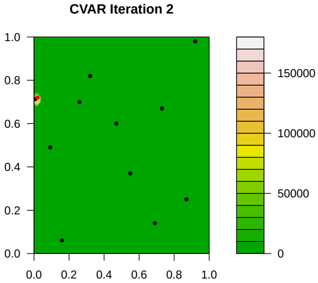


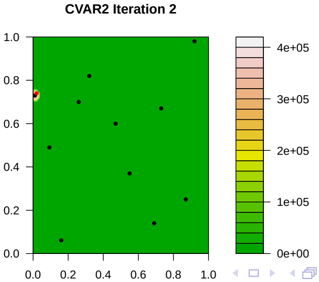
CVAR2 Iteration 1

# Illustrative 2d example

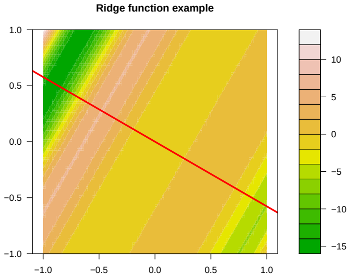

Ridge function example

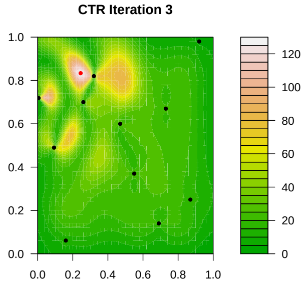

CTR Iteration 2

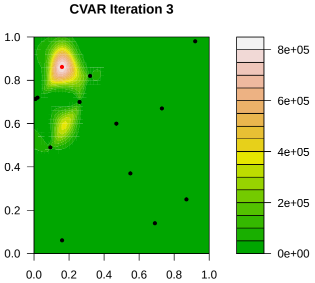

CVAR Iteration 2

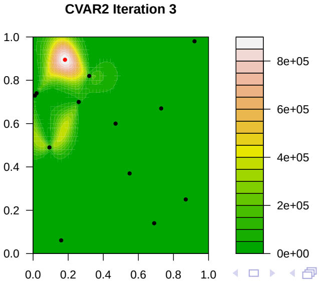

CVAR2 Iteration 2

# Illustrative 2d example

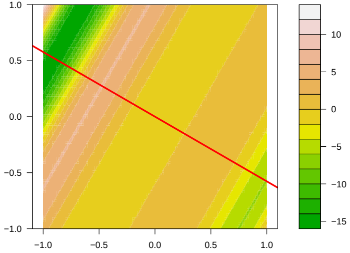

Ridge function example


CTR Iteration 3

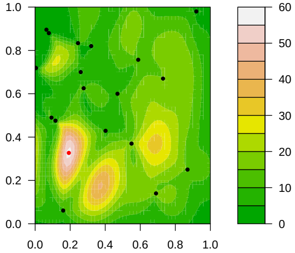

CVAR Iteration 3


CVAR2 Iteration 3

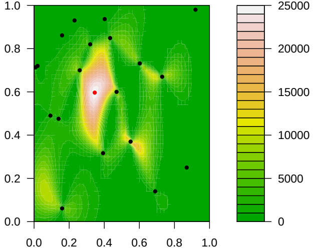
# Illustrative 2d example



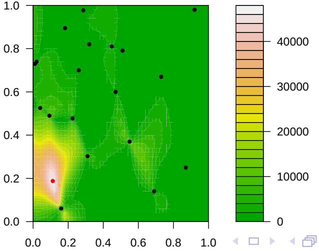
Ridge function example



CTR Iteration 10

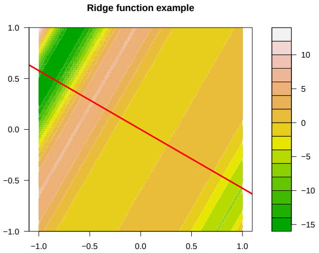


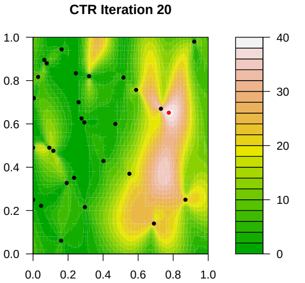
CVAR Iteration 10

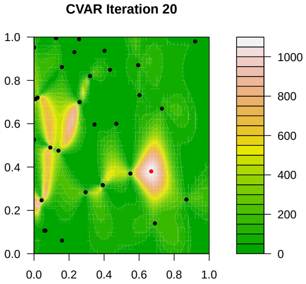


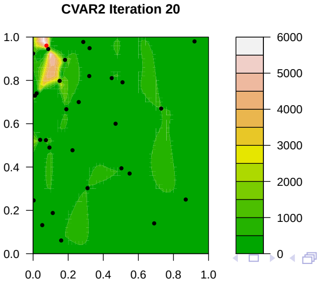
CVAR2 Iteration 10

# Illustrative 2d example


Ridge function example

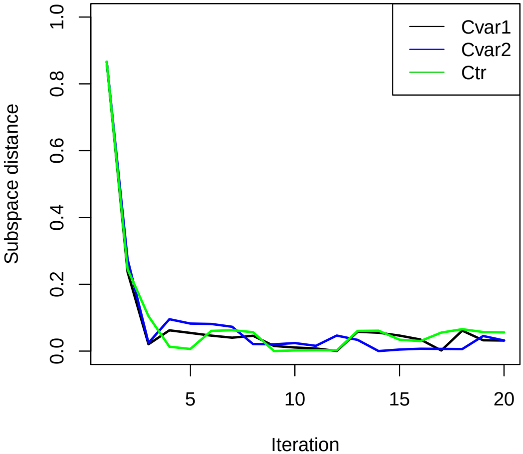

CTR Iteration 20


CVAR Iteration 20


CVAR2 Iteration 20

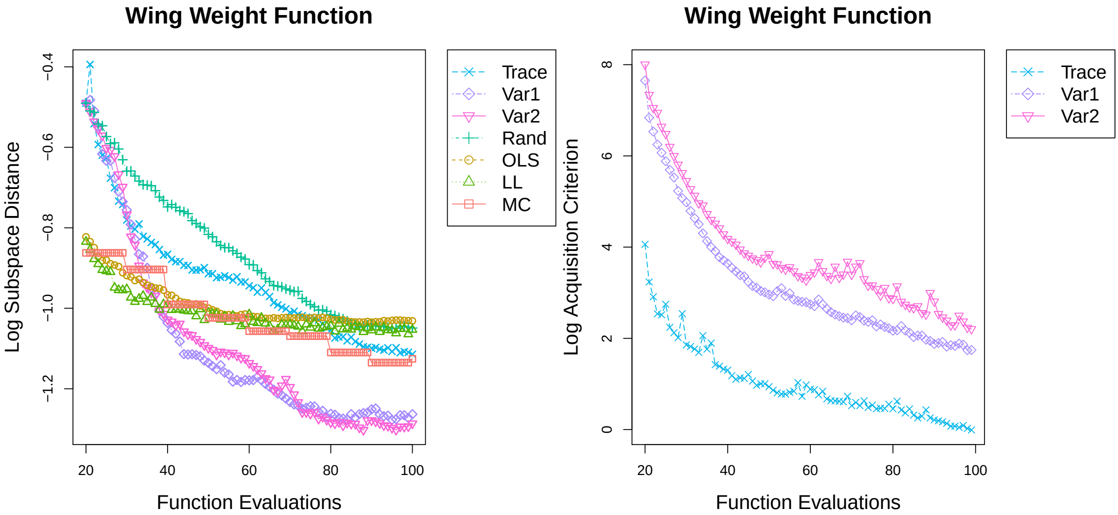
# Illustrative 2d example

Subspace distance: cosine of the first principle angle between the two subspaces



# Wing weight function

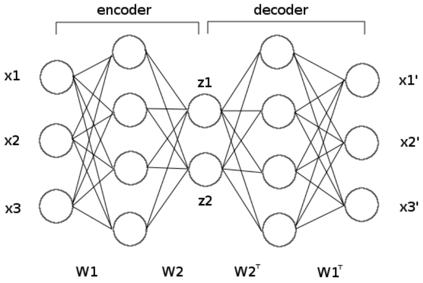
10 dimensional function, known to have a leading 1D active subspace.



# Non-linear embeddings

Linear dimension reduction may not be sufficient.

In this case, auto-encoders may become handy to learn a latent space: covariance kernel $k(\Psi(\mathbf{x}), \Psi(\mathbf{x}'))$ with $\Psi$ given by the encoder.



A convincing example[55] is with molecules, for which an efficient text representation exists (SMILES). Another popular model is the GP-LVM model[56].

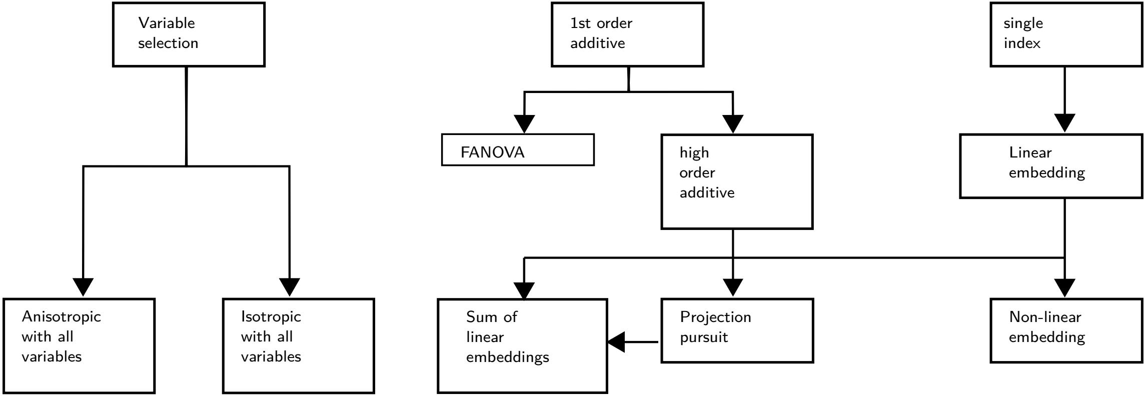
[55] R. Gómez-Bombarelli et al. “Automatic chemical design using a data-driven continuous representation of molecules”. In: *ACS central science* 4.2 (2018), pp. 268–276.

[56] N. Lawrence. “Probabilistic non-linear principal component analysis with Gaussian process latent variable models”. In: *The Journal of Machine Learning Research* 6 (2005), pp. 1783–1816.

# Summary

Most structural models are instances of the general one:

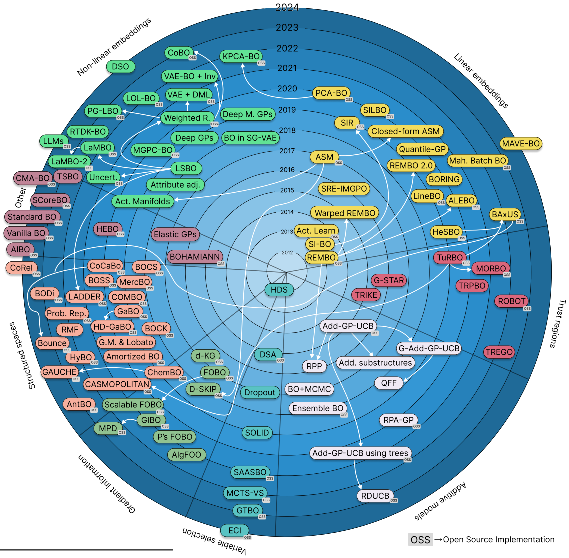
$$\text{model: } f(\mathbf{x}) \approx \sum_{i=1}^{\kappa} g_i(\mathbf{A}_i\mathbf{x})$$



For the last column, randomized directions are possible rather than needing full inference.

# Summary (cont'd)

More recent overview (figure borrowed from the paper):[57]



[57] M. González-Duque et al. "A survey and benchmark of high-dimensional Bayesian optimization of discrete sequences". In: *arXiv preprint arXiv:2406.04739* (2024).

# Outline

1. Background on Bayesian optimization
2. High-dimensional GPs
   - Overview
   - Additive models
   - Active subspace estimation
   - Other methods
3. Reconciling linear embedding and additive models?
4. Application to optimization
5. Conclusion

# Summary of pros and cons

## Pros

Additive models:

- interpretability
- keep the original variables
- simple orthogonality conditions

AS models:

- efficient dimension reduction
- often observed in practice
- capture high-order interactions

## Cons

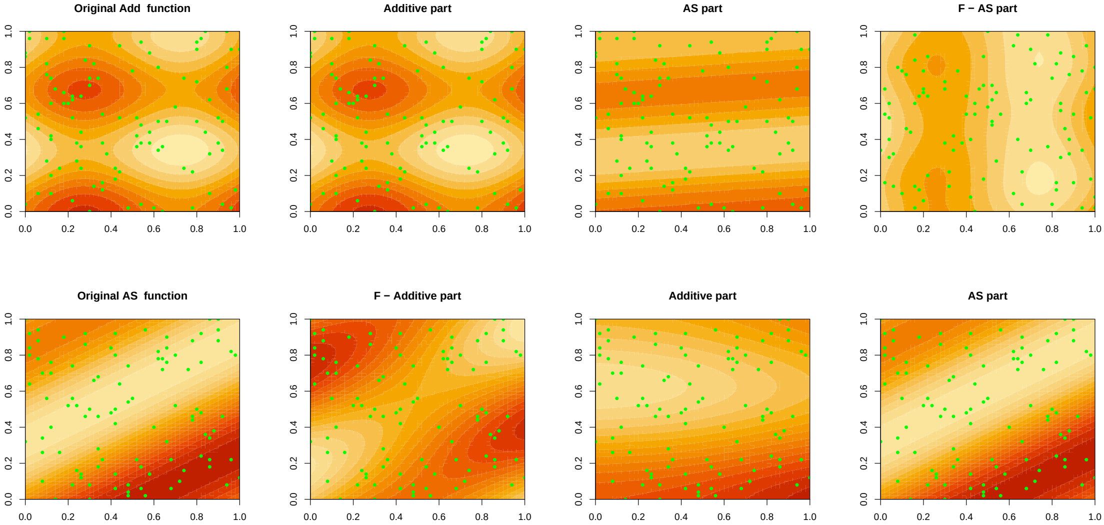
Additive models:

- inference is harder with increasing interaction order, or with complex dependency graphs
- difficult to learn high-order interaction terms

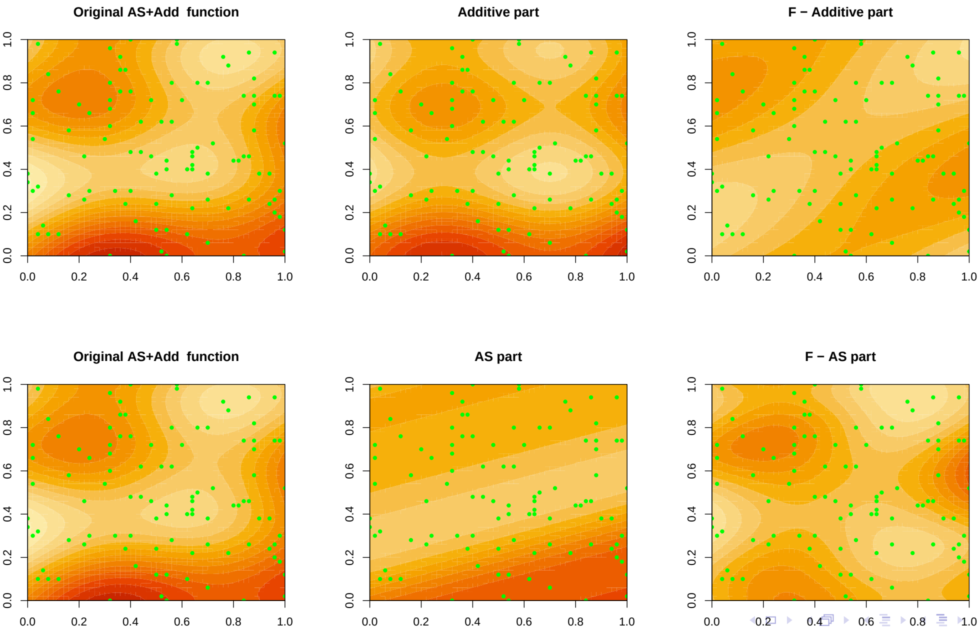
AS models:

- identifying the intrinsic dimension is complex
- for box-constraints, the resulting rotation is complexifying subsequent tasks

# Linear embedding versus additive model



# Linear embedding versus additive model (2)



Original AS+Add function

Additive part

F − Additive part

Original AS+Add function

AS part

F − AS part

# A multi-fidelity approach

Directly combining AS and additive models is difficult: there is an intersection between the features they capture.

Plus it is not clear how to enforce orthogonality à la[58]

Still, to benefit from both types of models, we propose[59] a less strict condition via an auto-regressive multi-fidelity approach[60]:

1. a first order additive model as the coarse model
2. an AS GP model as the fine level

This approach shows better results than a regular GP, with improved performance whenever additive and/or linear embedding structure is present.

[58] Lenz, *Additivity and Ortho-Additivity in Gaussian Random Fields*.

[59] Binois and Picheny, "Combining additivity and active subspaces for high-dimensional Gaussian process modeling".

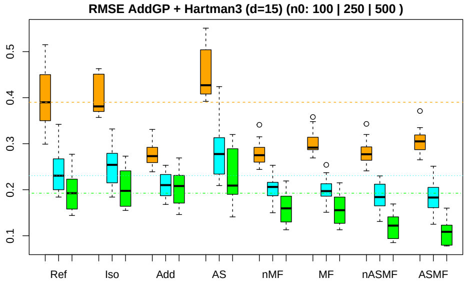
[60] M. C. Kennedy and A. O'Hagan. "Predicting the output from a complex computer code when fast approximations are available". In: *Biometrika* 87.1 (2000), pp. 1–13.

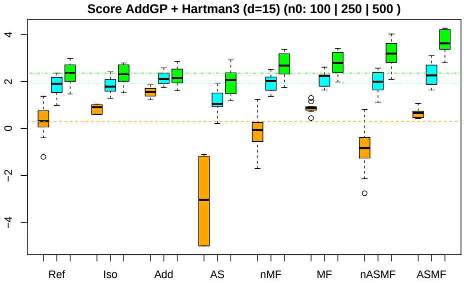
# Multi-fidelity approach summary

```
Input: $\mathbf{X}_E = \mathbf{X}_C$, $\mathbf{y}$, $p$ (e.g., 0.8)
Train an additive model $Y_C$ on $(\mathbf{X}_C, \mathbf{y})$
if $\tau_C^2 \leq 0.01 \times \sum_1^d \alpha_i$ then
    Sample $n_0 = p \times n$ data points from $\mathbf{x}_{1:n}$, $\mathbf{y}$ and remove the rest from $\mathbf{X}_C$ and $\mathbf{y}^{(C)}$.
    Train an additive model $Y_C$ on $(\mathbf{X}_C, \mathbf{y}^{(C)})$.
end if
Predict the response of $Y_C$ at $\mathbf{X}_E$: $m_n^{(C)}(\mathbf{X}_E)$.
Train a multi-fidelity GP from the residual data: $\mathbf{d} = \mathbf{y} - \rho m_n^{(C)}(\mathbf{X}_E)$.
Estimate the corresponding AS matrix $\mathbf{C}^{(n)}$.
Train an AS multi-fidelity GP, varying the number of dimensions kept $r$.
Output: Trained multi-fidelity model.
```

# MF approach example result

Test case: additive GP ($d = 15$) + rotated Hartmann3 function ($r = 3$) with varying budget (100, 250, 500 in orange, cyan, green)





# Outline

1. Background on Bayesian optimization
2. High-dimensional GPs
    - Overview
    - Additive models
    - Active subspace estimation
    - Other methods
3. Reconciling linear embedding and additive models?
4. Application to optimization
5. Conclusion

# BO is versatile ...

Active research directions on extensions include:

- batched versions of BO, e.g., with multi-point EI or local models
- noise on inputs/outputs, heteroskedasticity, non-Gaussian noise
- complex inputs/outputs (images, graphs, functions, ...)
- modeling non-stationarity
- multi-fidelity and variable cost
- multi/many objective, multi-task, constrained optimization

Review papers:[61],[62]
Books:[63],[64]

[61] B. Shahriari et al. "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175.

[62] P. Frazier. "Bayesian Optimization". INFORMS Tutorials. 2018.

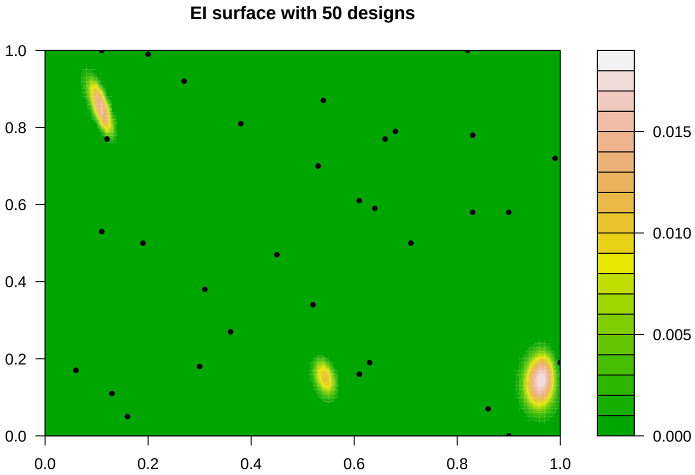
[63] R. B. Gramacy. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. CRC Press, 2020.

[64] Garnett, *Bayesian Optimization*.

# … with some practical limitations

1) GP training is expensive: the vanilla version is $\mathcal{O}(n^3)$ in time complexity (but can be reduced to $\mathcal{O}(n)$ with approximations).

2) Optimizing EI (or other) is increasingly hard as n grows.

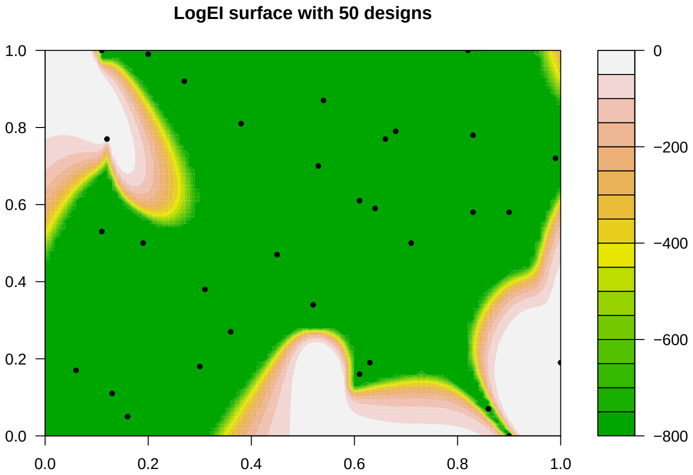


3) High dimension exacerbates these effects.

# … with some practical limitations

1) GP training is expensive: the vanilla version is $\mathcal{O}(n^3)$ in time complexity (but can be reduced to $\mathcal{O}(n)$ with approximations).

2) Optimizing EI (or other) is increasingly hard as n grows.



3) High dimension exacerbates these effects.

# General notes

BO failures not only come from a bad surrogate, but also from difficult acquisition function optimization.

Some recent ideas:

- use log $EI$ rather than EI (plus some more stable expressions)[65];
- use of more powerful optimizers
  - compositional (nested expectation, e.g., $\mathbf{E}_{z \sim \mathcal{N}(0,I)}\left[\max_j(a_j + B_j z)\right]$) ones[66],
  - composite ($\max g \circ f$)[67],
  - even just gradient-based ones with auto-differentiation.

[65] S. Ament et al. "Unexpected improvements to expected improvement for Bayesian optimization". In: *NeurIPS* 36 (2024).

[66] A. Grosnit et al. "Are we forgetting about compositional optimisers in Bayesian optimisation?" In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 7183–7260.

[67] J. Larson and M. Menickelly. "Structure-aware methods for expensive derivative-free nonsmooth composite optimization". In: *Mathematical Programming Computation* 16.1 (2024), pp. 1–36.

# Random or not random?

Compared to a given sample analysis, where getting the best possible model is important, the sequential aspect of BO enables more strategies:

- balancing learning the structure and optimization (more on this later for AS);
- randomize the model structure parameters:
  - with random embedding decompositions[68]
  - with random 1d projection[69];
  - with random additive decomposition[70],[71];

[68] Z. Wang et al. "Bayesian Optimization in a Billion Dimensions via Random Embeddings". In: *Proceedings of IJCAI* (2013); Z. Wang et al. "Bayesian Optimization in a Billion Dimensions via Random Embeddings". In: *Journal of Artificial Intelligence Research (JAIR)* 55 (2016), pp. 361–387.

[69] J. Kirschner et al. "Adaptive and Safe Bayesian Optimization in High Dimensions via One-Dimensional Subspaces". In: *International Conference on Machine Learning*. 2019, pp. 3429–3438.

[70] Wang et al., "Batched Large-scale Bayesian Optimization in High-dimensional Spaces".

[71] Ziomek and Ammar, "Are random decompositions all we need in high dimensional Bayesian optimisation?"

# Benchmarking high-dimensional BO

There are not so many exhaustive benchmark results available, except, e.g.,[72], plus results in publications are sometimes contradictory.

Reasons may include:

- Publication pressure;
- Package defaults may not be suited to high-dimensional BO;
- Complex models are more prone to instability;
- Codes may not be available or not with a simple interface;
- Spread between several research communities and package tools;
- Computational cost is even larger than usual;
- Tests in the literature are with different budgets and dimensions.
- What are good benchmark functions? Are there realistic high dim examples?

[72] M. L. Santoni et al. "Comparison of high-dimensional Bayesian optimization algorithms on bbob". In: *ACM Transactions on Evolutionary Learning* 4.3 (2024), pp. 1–33.

# Trust region based BO, e.g.,[73],[74]

The main idea is to focus on a ball centered on the best design, whose radius is:

- decreased if the search is unsuccessful;
- increased otherwise.

It helps reducing the over-exploration issue of most infill criteria when $d$ is large.

Furthermore, the acquisition function optimization is restricted to a few directions at once.

Local GPs may be used, to help with non-stationarity.

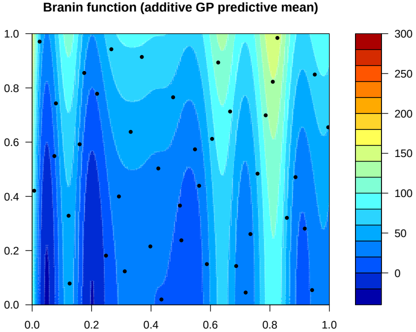
[73] Y. Diouane et al. “TREGO: a trust-region framework for efficient global optimization”. In: *Journal of Global Optimization* 86.1 (2023), pp. 1–23.

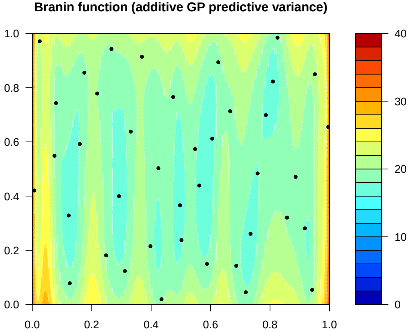
[74] D. Eriksson and M. Poloczek. “Scalable constrained Bayesian optimization”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 730–738.

# Optimizing with additive structure

Typically, the acquisition function follows the same decomposition as the additive GP: $g_i(\mathbf{x}_I) = \mathcal{N}(m_{n,i}(\mathbf{x}_I), s^2_{n,i}(\mathbf{x}_I))$ where $s^2_{n,I}(\mathbf{x}_I) = k_I(\mathbf{x}_I, \mathbf{x}_I) - \mathbf{k}_I(\mathbf{x}_I)^\top \mathbf{K}^{-1} \mathbf{k}_I(\mathbf{x}_I)$ for a general index $I$.
$\rightarrow$ this simplifies its optimization and allows message passing optimization, e.g.,[75].

The main drawback is that the variance may be zero at unobserved locations (but noise is usually added).


Branin function (additive GP predictive mean)


Branin function (additive GP predictive variance)

[75] Rolland et al., "High-Dimensional Bayesian Optimization via Additive Models with Overlapping Groups".

# Screening: domain issues with dimension reduction

Whenever optimizing in a reduced search space, one question is how to choose the remaining values.

Possible strategies include:

- using an arbitrary value;
- using some prior knowledge;
- using the values from the best design so far;
- using random values.

These effects are exacerbated with more complex dimension reduction.

# Optimizing with an active subspace[76]

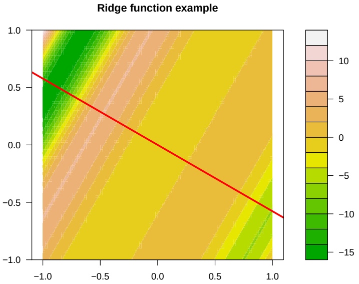
Let $\mathbf{W} = [\mathbf{A}\ \mathbf{W}_2]$ be a basis of $\mathbb{R}^d$.

Splitting between active and inactive variables:
$\forall \mathbf{x} \in \mathbb{R}^d = \mathbf{W}\mathbf{W}^\top \mathbf{x} = \mathbf{A}\mathbf{A}^\top \mathbf{x} + \mathbf{W}_2\mathbf{W}_2^\top \mathbf{x} = \mathbf{A}\mathbf{y} + \mathbf{W}_2\mathbf{z}$, $\mathbf{y} \in \mathbb{R}^d$, $\mathbf{z} \in \mathbb{R}^{d-r}$

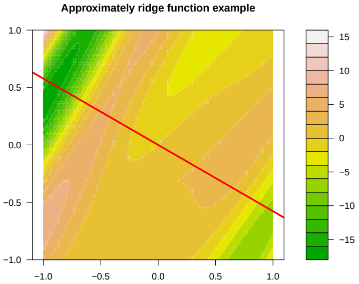
If $f$ has a true active subspace: find $\mathbf{y}^* \in \operatorname*{argmin}_{y \in \mathbb{R}^d} f(\mathbf{A}\mathbf{y})$

Else, the problem is: find $\mathbf{y}^* \in \operatorname*{argmin}_{y \in \mathbb{R}^d} \min_{\mathbf{z} \in \mathbb{R}^{d-r}} f(\mathbf{A}\mathbf{y} + \mathbf{W}_2\mathbf{z})$

Both are more complex for a compact domain $\mathbf{X}$.



Ridge function example

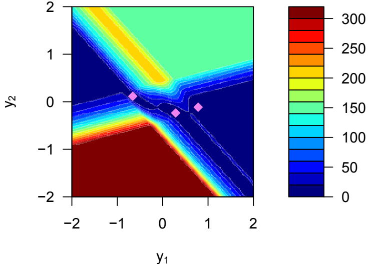


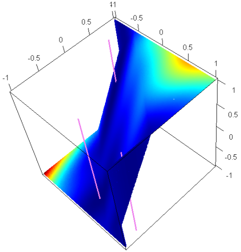
Approximately ridge function example

[76] Constantine, *Active subspaces: Emerging ideas for dimension reduction in parameter studies.*

# Random embeddings / Active subspace domain issue

Domain issues have been studied mostly from the random embedding point of view, starting with REMBO[77], and further works[78],[79],[80].





Convergence results depend on both **A** and the low-dimensional search space. Recent theoretical results come from global optimization[81].

[77] Wang et al., "Bayesian Optimization in a Billion Dimensions via Random Embeddings".

[78] M. Binois, D. Ginsbourger, and O. Roustant. "On the choice of the low-dimensional domain for global optimization via random embeddings". In: *Journal of global optimization* 76.1 (2020), pp. 69–90.

[79] A. Nayebi, A. Munteanu, and M. Poloczek. "A Framework for Bayesian Optimization in Embedded Subspaces". In: *ICML*. 2019, pp. 4752–4761.

[80] B. Letham et al. "Re-Examining Linear Embeddings for High-Dimensional Bayesian Optimization". In: *NeurIPS*. ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1546–1558.

[81] C. Cartis, E. Massart, and A. Otemissov. "Bound-constrained global optimization of functions with low effective dimensionality using multiple random embeddings". In: *Mathematical Programming* 198.1 (2023), pp. 997–1058.

# Tradeoff between optimization and dimension reduction

These two tasks are possibly conflicting, hence benefiting from a multi-objective point of view.

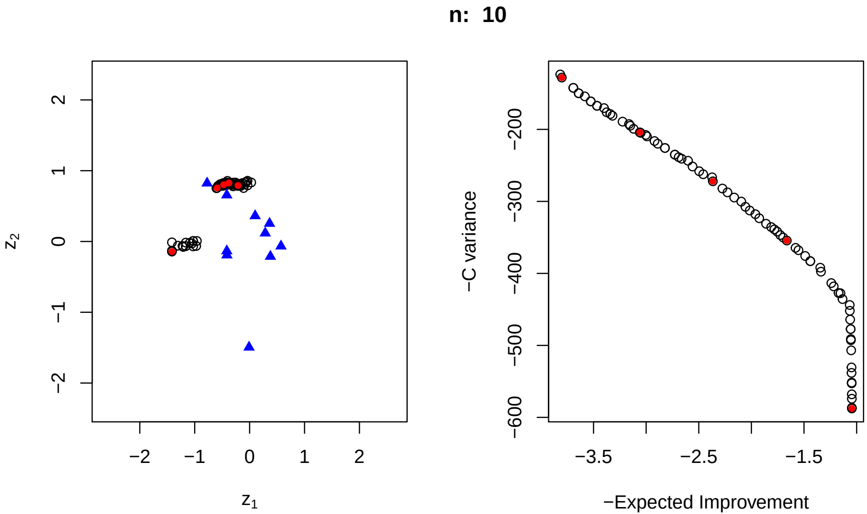
Let's thus estimate the Pareto front between Expected Improvement and active subspace variance.

### Pseudo code for BO with active subspace learning

```
Require: $n_0$, $r$.
Construct an initial design of experiments in $\mathcal{X}$, of size $n_0$.
Build the (high-dimensional) GP model with kernel $k$.
while time/evaluation budget not exhausted do
    Compute the active subspace matrix $\mathbf{C}^{(n)}$ and $\mathbf{A}^{(n)}$ (rank r)
    Construct a low dimensional GP based on $\mathbf{A}^{(n)}$
    Find $\mathbf{z}^* \in \underset{\mathcal{Z}=\mathbf{A}^{(n)}\mathbf{X}}{\operatorname{argmin}} \left(-EI(\mathbf{z}), -J(\mathbf{z})\right)$
    Evaluate the objective function
    Update the high-dimensional GP model based on new data.
end while
```

# Illustration on the wing weight function

10 dimensional function, known to have a leading 1D active subspace.
5 points are selected from the Pareto front at each iteration.
Total budget: 50, $r = 2$, $n_0 = 10$

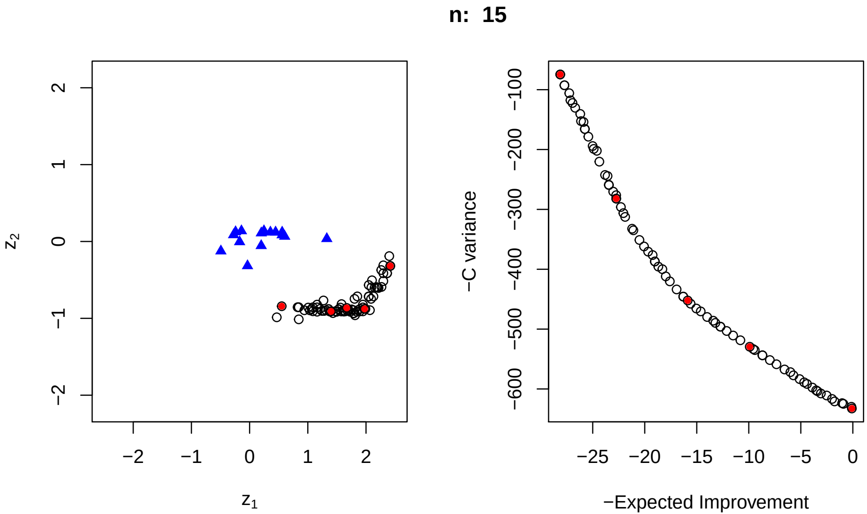


# Illustration on the wing weight function

10 dimensional function, known to have a leading 1D active subspace.

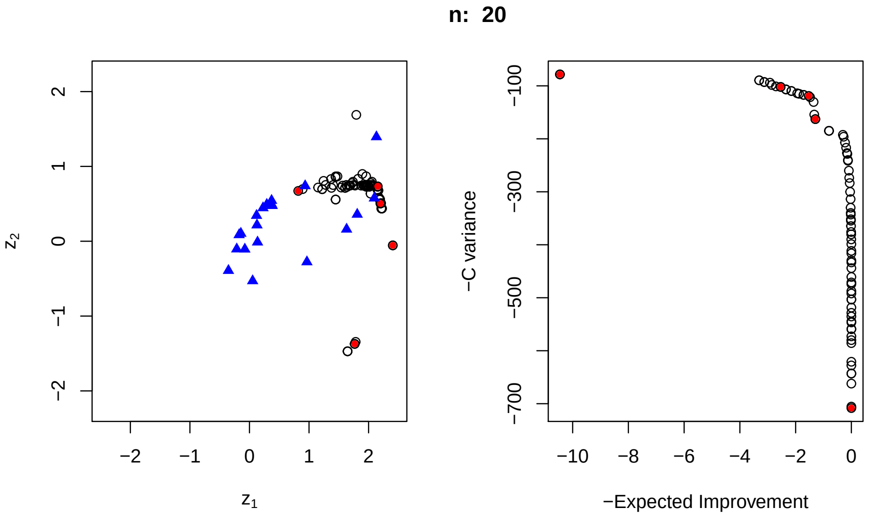
5 points are selected from the Pareto front at each iteration.

Total budget: 50, $r = 2$, $n_0 = 10$



# Illustration on the wing weight function

10 dimensional function, known to have a leading 1D active subspace.
5 points are selected from the Pareto front at each iteration.
Total budget: 50, $r = 2$, $n_0 = 10$

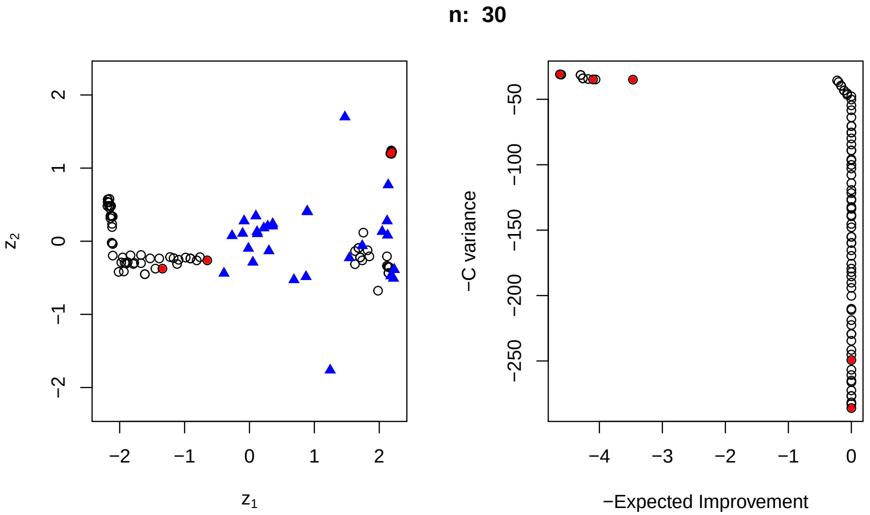


# Illustration on the wing weight function

10 dimensional function, known to have a leading 1D active subspace.

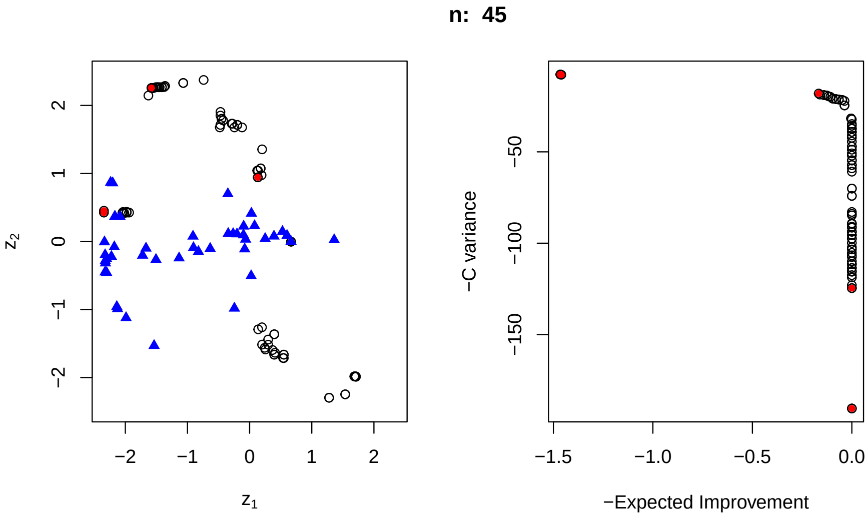
5 points are selected from the Pareto front at each iteration.

Total budget: 50, $r = 2$, $n_0 = 10$



# Illustration on the wing weight function

10 dimensional function, known to have a leading 1D active subspace.
5 points are selected from the Pareto front at each iteration.
Total budget: 50, $r = 2$, $n_0 = 10$

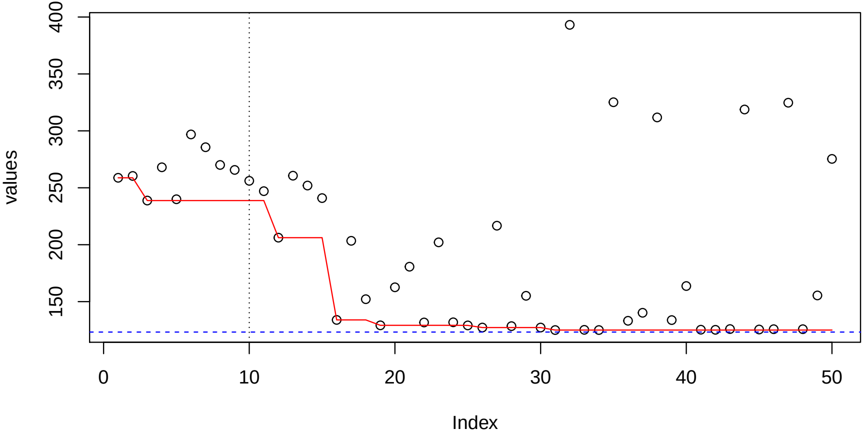


# Illustration on the wing weight function

10 dimensional function, known to have a leading 1D active subspace.

5 points are selected from the Pareto front at each iteration.

Total budget: 50, $r = 2$, $n_0 = 10$



# Outline

1. Background on Bayesian optimization
2. High-dimensional GPs
   - Overview
   - Additive models
   - Active subspace estimation
   - Other methods
3. Reconciling linear embedding and additive models?
4. Application to optimization
5. Conclusion

# Conclusion and perspectives

High-dimensional GP modeling is a compromise between:

- prior knowledge;
- accuracy (related to the budget $n$);
- inference complexity, randomization, orthogonality;
- interpretability;
- the task at end.

For optimization, additional challenges are

- to learn what is important for low values;
- the low-dimensional search space (if applicable);
- the ability to recover from a wrong structure;
- the interplay between global and local aspects.

Many opportunities in hybridizing

Thank you for your attention