Ansys / DYNARDO
presented at the 17th Weimar Optimization and Stochastic Days 2020 | Source: www.dynardo.de/en/library

PROBABILISTIC INTELLIGENCE

Efficient Bayesian Optimization for high-dimensional problems

Machine Learning based on Probabilistic Intelligence

Kevin Cremanns[1, 2], Prof. Dirk Roos[2]

25.06.2020

[1]PI Probaligence GmbH
[2]Niederrhein University of Applied Sciences - IMH Institute of Modelling and High-Performance Computing

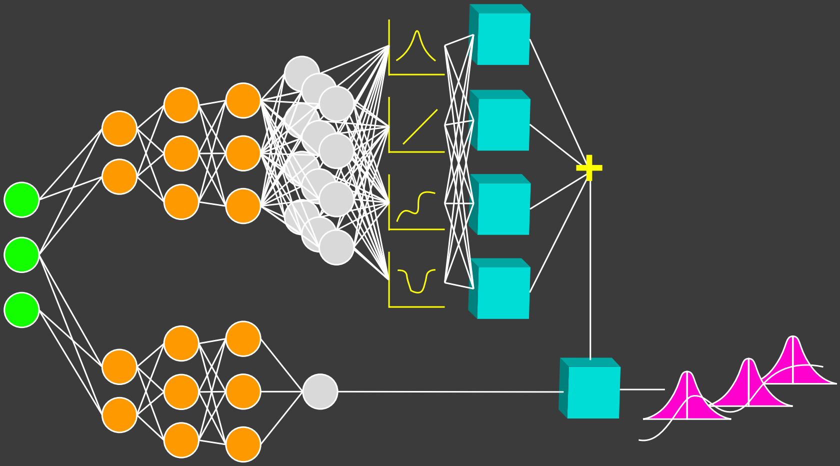
Hochschule Niederrhein
University of Applied Sciences

IMH
Institut für Modellbildung und Hochleistungsrechnen
Institute of Modelling and High-Performance Computing

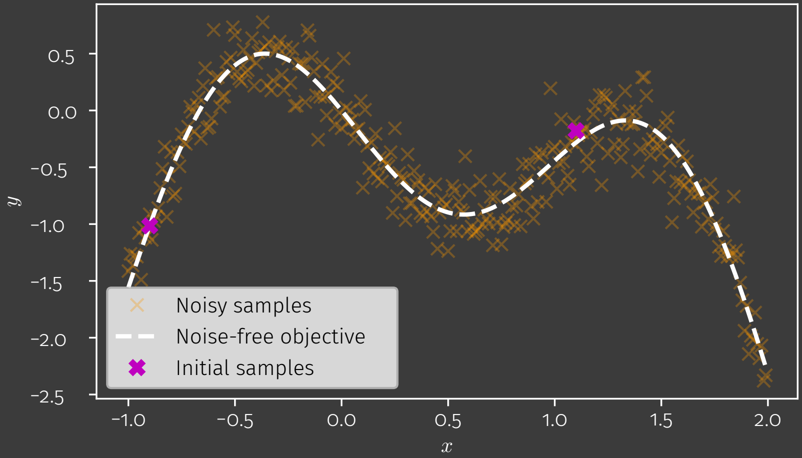
Bayesian optimization

Deep Infinite Mixture of Gaussian Processes - $DIM - \mathcal{GP}$

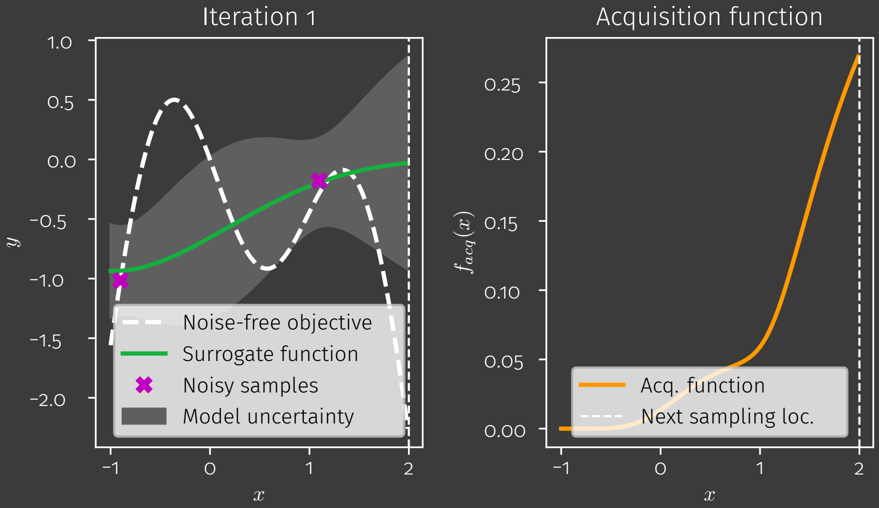
Unique combination of Deep Neural Networks and Gaussian Processes:



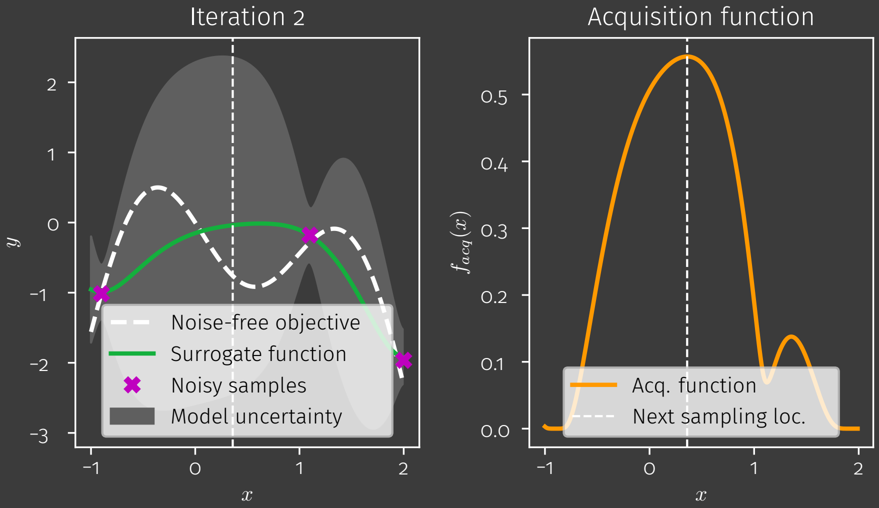
Bayesian optimization - introduction



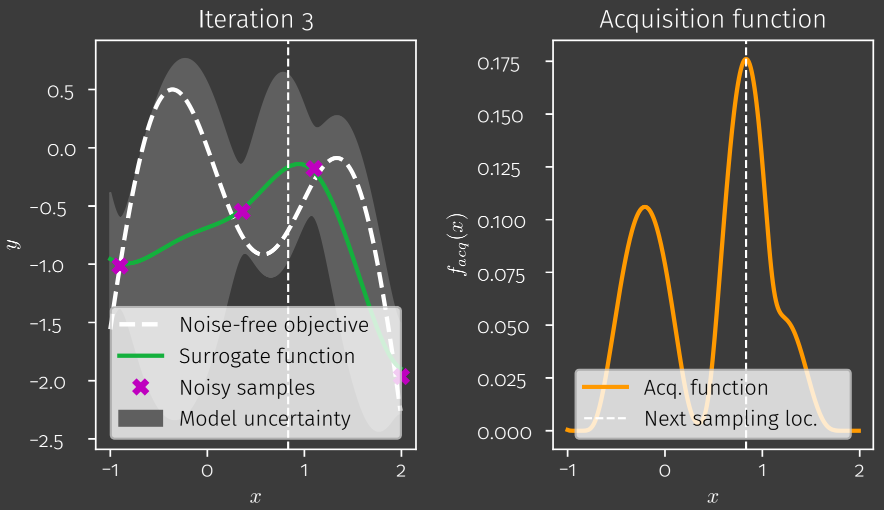
Bayesian optimization - introduction



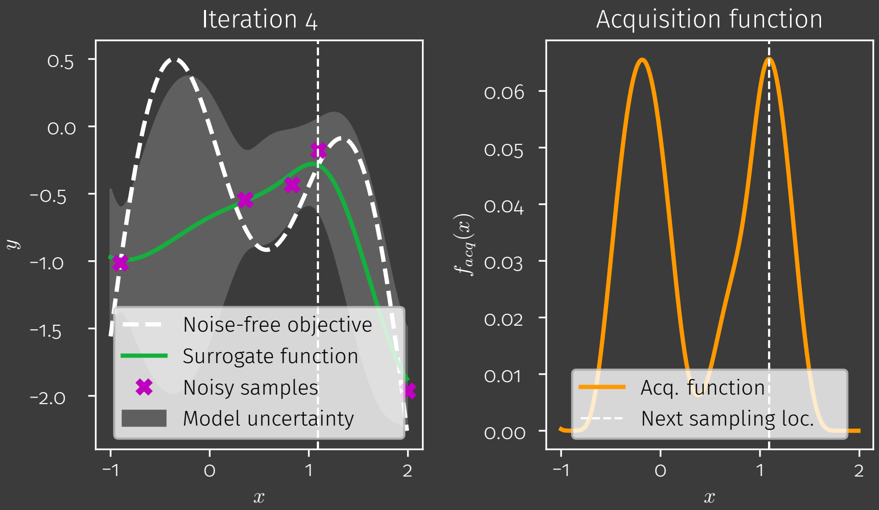
Bayesian optimization - introduction



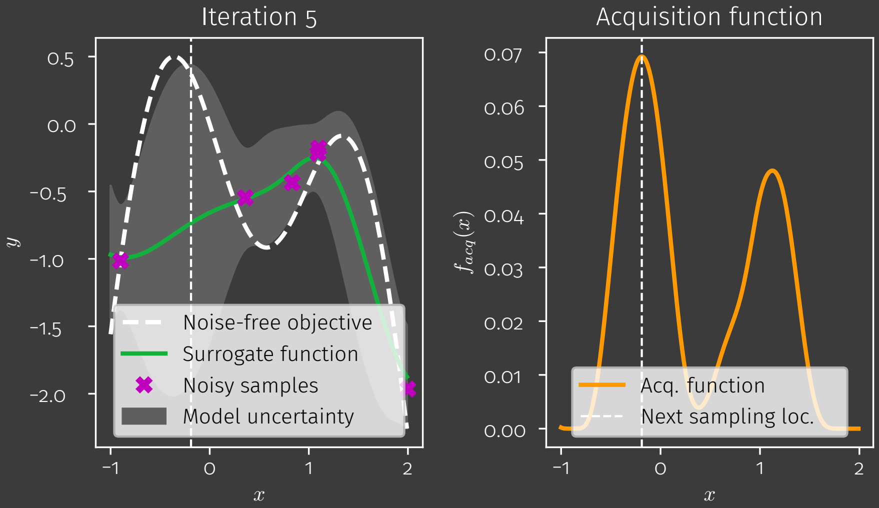
Bayesian optimization - introduction



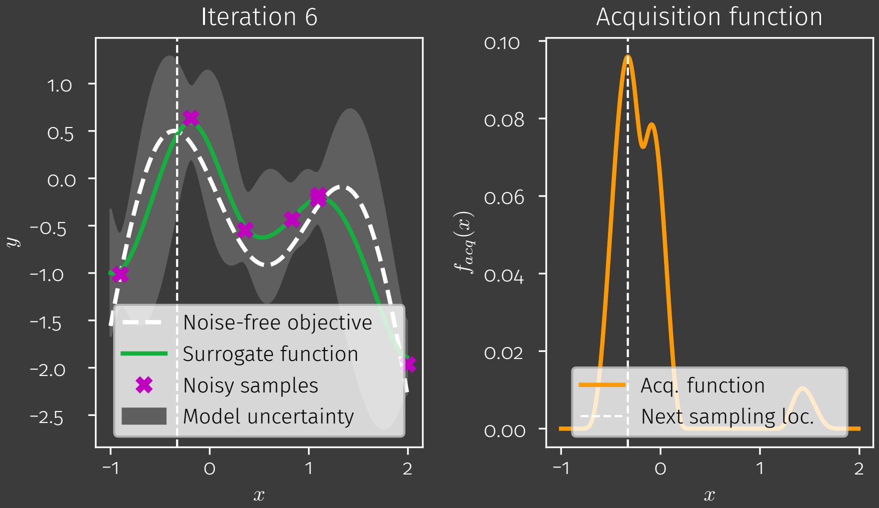
Bayesian optimization - introduction



Bayesian optimization - introduction

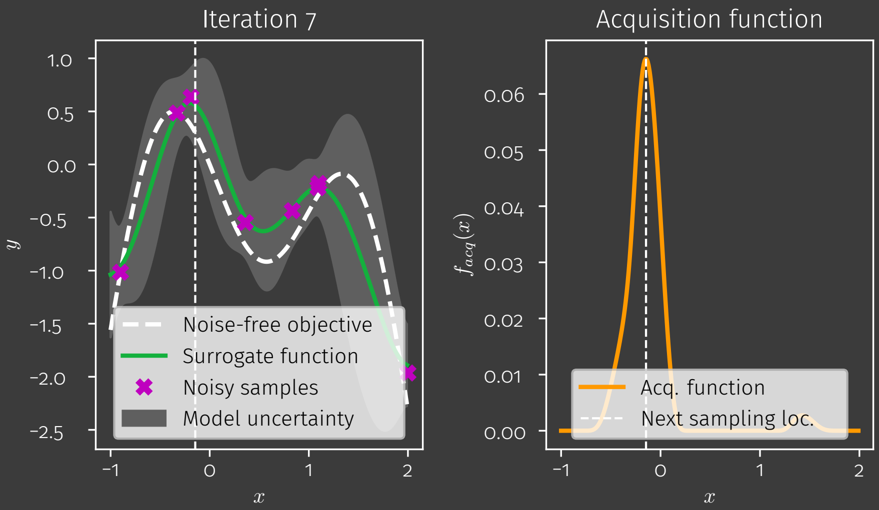


Bayesian optimization - introduction

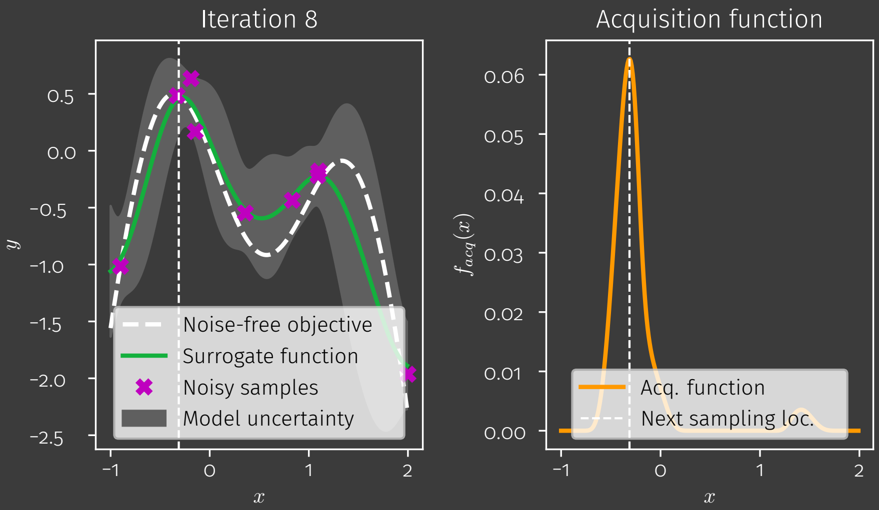


π^2

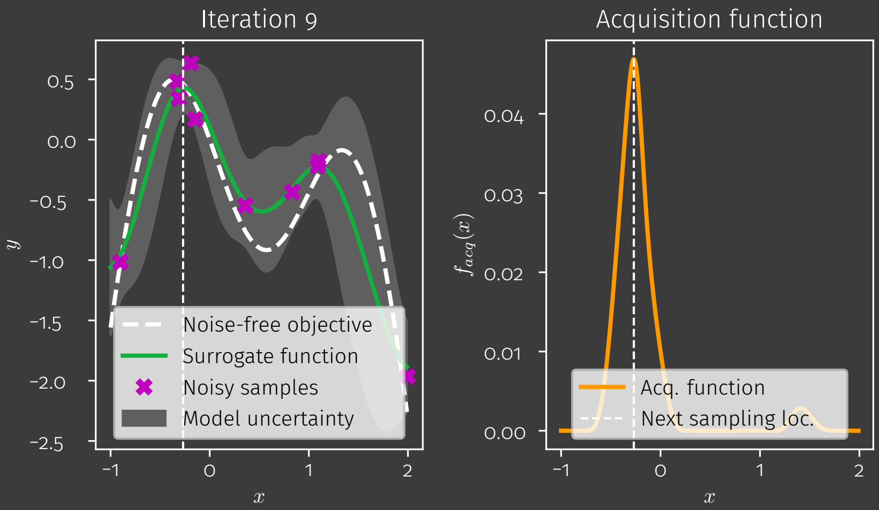
Bayesian optimization - introduction



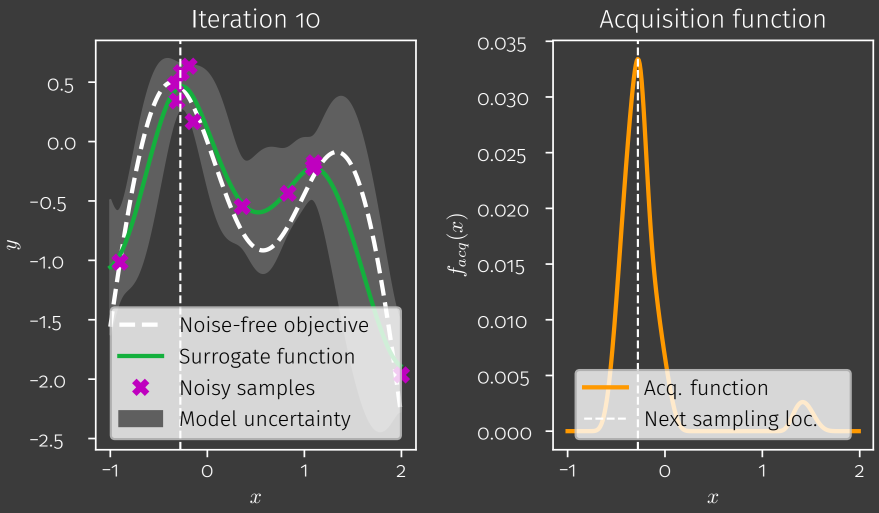
Bayesian optimization - introduction



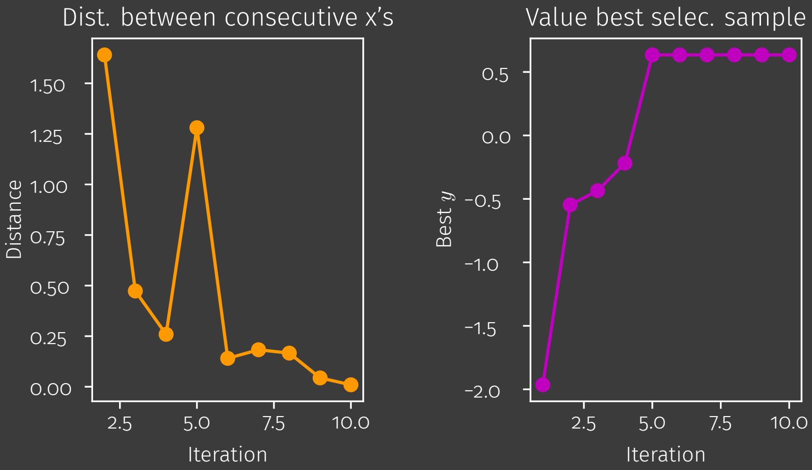
Bayesian optimization - introduction



Bayesian optimization - introduction



Bayesian optimization - introduction



High-dimensional optimization (1/2)

Why is high-dimensional optimization difficult?

- The search space grows exponentially as the number of dimensions increases. Therefore the curse of dimensionality makes it difficult to lead effectively the optimization process.
- The properties of the search space may change as the dimensionality grows up, e.g., multi-modal functions.
- The evaluation of solutions demands increasing computational resources and runtime.
- ...

High-dimensional optimization (2/2)

- If gradients for the objectives / constraints are available, a gradient based optimization algorithm like L-BFGS-B might be a solution.
- Typically, many real-world optimization problems have no gradients available and / or the objective and constraints are discrete or non-differentiable.
- Therefore, nature inspired based algorithms like PSO or evolutionary strategies must be used but they have especially problems with high-dimensional tasks.

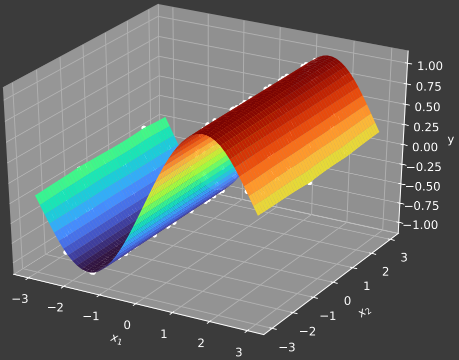
High-dimensional optimization (2/2)

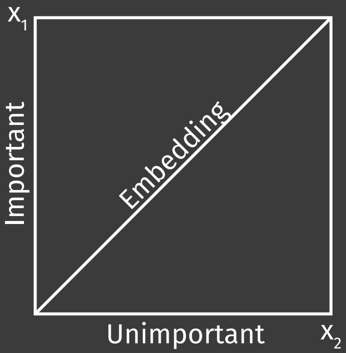
- If gradients for the objectives / constraints are available, a gradient based optimization algorithm like L-BFGS-B might be a solution.
- Typically, many real-world optimization problems have no gradients available and / or the objective and constraints are discrete or non-differentiable.
- Therefore, nature inspired based algorithms like PSO or evolutionary strategies must be used but they have especially problems with high-dimensional tasks.

So what to do?

Sobol Indice based eMbedding Bayesian Optimization - SIMBO

REMBO - Random EMbedding Bayesian Optimization





Optimization in random linear embeddings is easier and faster especially for high dimensional problems.

Embedding

Original design matrix ($D = 3$, $N = 3$, x_1, x_2, x_3):

$$X_o^{[D \times N]} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Embedding matrix ($d = 2$):

$$A^{[d \times D]} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{pmatrix}$$

Embedded design matrix:

$$X_e^{[d \times N]} = AX_o$$

$$X_e = \begin{pmatrix} 12 & 15 & 18 \\ 24 & 30 & 36 \end{pmatrix}$$

D: orig. dim., d: embd dim., N: Num. samples

REMBO - Advantages & disadvantage

Advantages:

- Increases performance for high dimensional optimization problems with unimportant optimization parameters
- Speeds up optimization calculations

Disadvantages:

- Correct choice of the embedding dimensionality d
- Random choice of the embedding matrix A can be unlucky

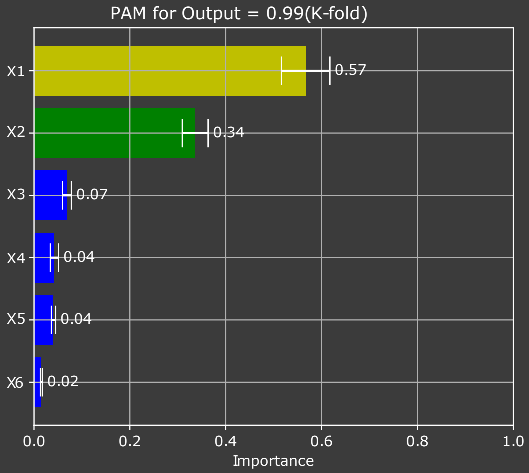
REMBO

How to choose embedding matrix A and embedding dim. d ?

Sobol-indice-based sensitivity analysis

$$S_i = \frac{Var_{X_i}(E_{X_{\sim i}}(\boldsymbol{y}|x_i))}{Var(\boldsymbol{y})};\ S_{T_i} = \frac{E_{X_{\sim i}}(Var_{X_i}(\boldsymbol{y}|X_{\sim i}))}{Var(\boldsymbol{y})}$$

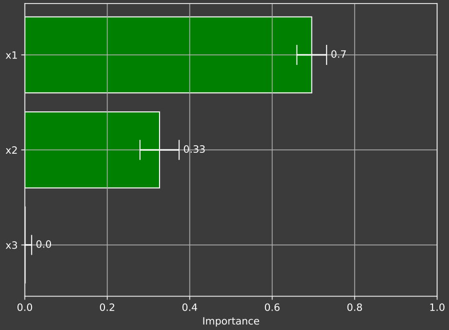
- If non-additive interaction effects between x, y are present $\sum_i^n S_{Ti} > 1$.
- or if $\sum_i^n S_{Ti} = 1$, then pure additive effects exist.
- Non-linear, non-monotonic, multivariate sensitivities.
- Estimation of confidence intervals.



SIEMBO - Example for A - $N = 3$

Simple example:

- $D = 3$ with x_1, x_2, x_3:
- $y = x_1 2 + x_2$
- x_3 unimportant
- $d = 3$ cause we don't know better

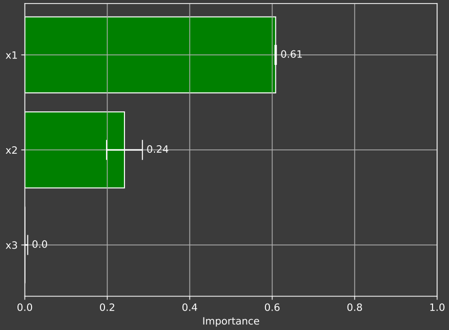


$$A^{[d \times D]} = \begin{pmatrix} \mathcal{N}(0.7, 0.04^2) & \mathcal{N}(0, 1^2) & \mathcal{N}(0, 1^2) \\ \mathcal{N}(0, 1^2) & \mathcal{N}(0.33, 0.05^2) & \mathcal{N}(0, 1^2) \\ \mathcal{N}(0, 1^2) & \mathcal{N}(0, 1^2) & \mathcal{N}(0, 0.01^2) \end{pmatrix}$$

SIEMBO - Example for A - $N = 6$

Simple example:

- $D = 3$ with x_1, x_2, x_3:
- $y = x_1 2 + x_2$
- x_3 unimportant
- $d = 3$ set automatically, all inputs have a chance to be important

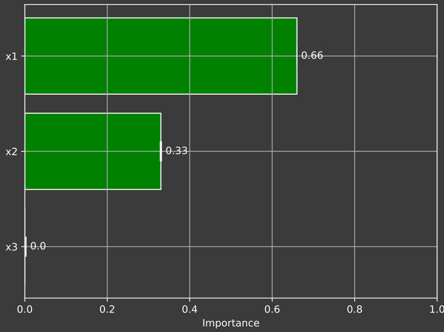


$$A^{[d \times D]} = \begin{pmatrix} \mathcal{N}(0.61, 0.01^2) & \mathcal{N}(0, 1^2) & \mathcal{N}(0, 1^2) \\ \mathcal{N}(0, 1^2) & \mathcal{N}(0.24, 0.04^2) & \mathcal{N}(0, 0.01^2) \\ \mathcal{N}(0, 1^2) & \mathcal{N}(0, 1^2) & \mathcal{N}(0, 0.01^2) \end{pmatrix}$$

SIEMBO - Example for A - $N = 12$

Simple example:

- $D = 3$ with x_1, x_2, x_3:
- $y = x_1 2 + x_2$
- x_3 unimportant
- $d = 2$ set automatically, x_3 unimportant



$$A^{[d \times D]} = \begin{pmatrix} \mathcal{N}(0.66, 0^2) & \mathcal{N}(0, 1^2) & \mathcal{N}(0, 1^2) \\ \mathcal{N}(0, 1^2) & \mathcal{N}(0.33, 0^2) & \mathcal{N}(0, 1^2) \end{pmatrix}$$

SIEMBO - Sobol Indice EMbedding Bayesian Optimization

Advantages:

- Increases performance for high dimensional optimization problems with unimportant optimization parameters
- Speeds up optimization calculations
- Builds linear combination of the most important parameters
- The embedding dimensionality d will be selected / adapted automatically during each optimization step

Disadvantages:

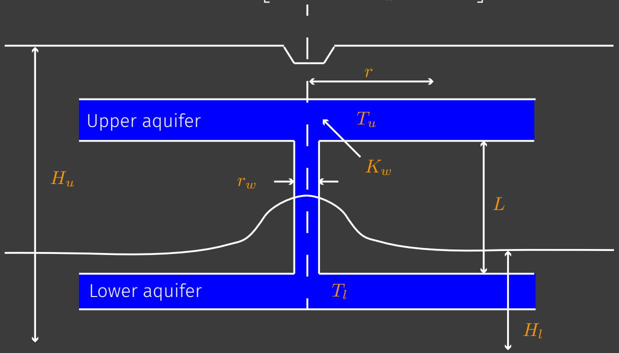
- Calculation of the Sobol indices for each optimization iteration
- User must set a upper bound for d or set $d = D$ and let it adjust automatically

Applications

Borehole optimization

Maximize water flow rate:

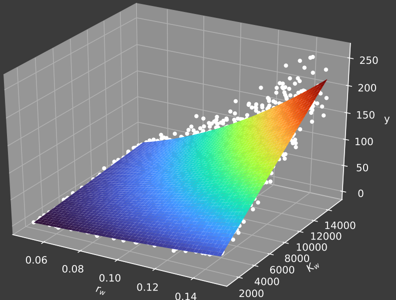
$$y = \frac{2\pi T_u (H_u - H_l)}{\ln(r/r_w)\left[1 + \frac{2 L T_u}{ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_l}\right]}$$



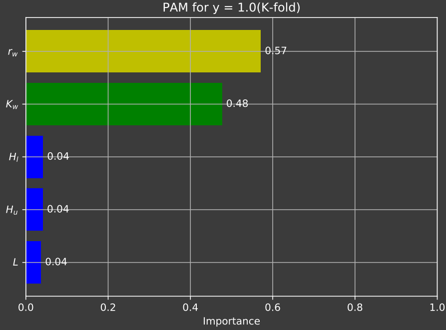
- y [0, 385] water flow rate $[m^3/yr]$
- $r_w \in [0.05, 0.15]$ radius of borehole $[m]$
- $r \in$ [100, 50 000 radius of influence $[m]$
- $T_u \in$ [63 070, 115 600] transmissivity of upper aquifer $[m^2/yr]$
- $H_u \in$ [990, 1 110] potentiometric head of upper aquifer $[m]$
- $T_l \in$ [63.1, 116] transmissivity of lower aquifer $[m^2/yr]$
- $H_u \in$ [700, 820] potentiometric head of lower aquifer $[m]$
- $L \in$ [1 120, 11 680] length of borehole $[m]$
- $K_w \in$ [9 855, 12 045] hydraulic conductivity of borehole $[m/yr]$

Borehole optimization

Surface plot:



Sensitivity:



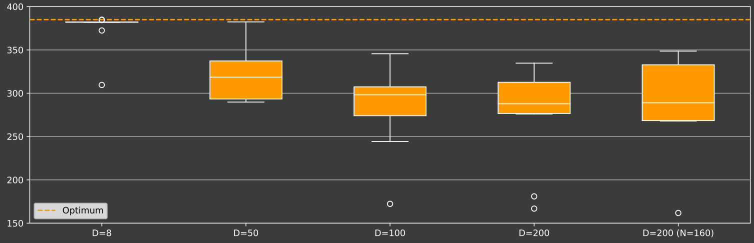
Borehole optimization

Optimization setup:

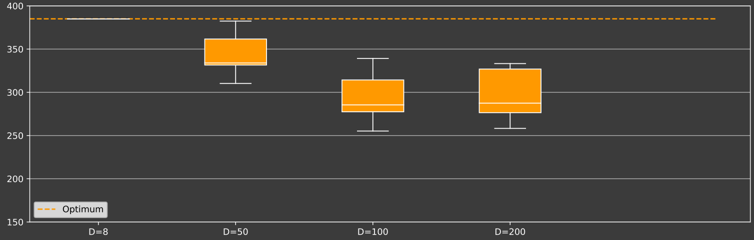
- Add unimportant parameters $[0, 42, 92, 192]$, so $D = 8$, $D = 50$, $D = 100$, $D = 200$ with $d = 8$
- 10 start samples and 50 adaptation steps with 1 design per adaptation
- 10 repetitions of each optimization
- Comparison of Stochos without SIEMBO, Stochos with SIEMBO and OptiSLang AMOP
- AMOP Settings: All models and 100% optimization criteria

Borehole optimization - objective comparison

OptiSLang AMOP:

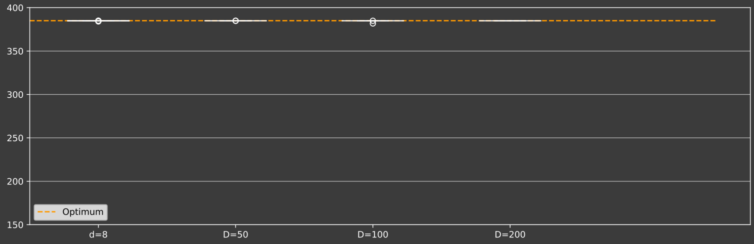


Stochos without SIEMBO:



Borehole optimization - objective comparison

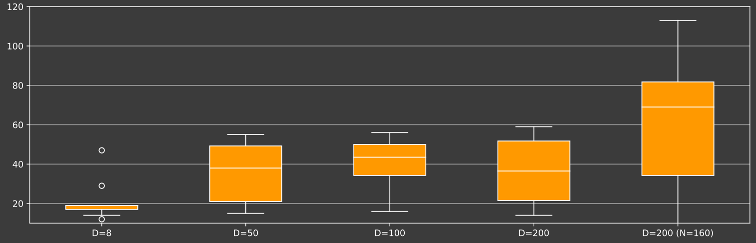
Stochos with SIEMBO:



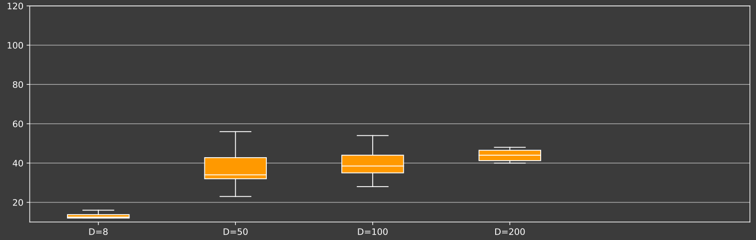
- Constantly achieving the optimum regardless of the number of dimensions

Borehole optimization - sample comparison

OptiSLang AMOP:

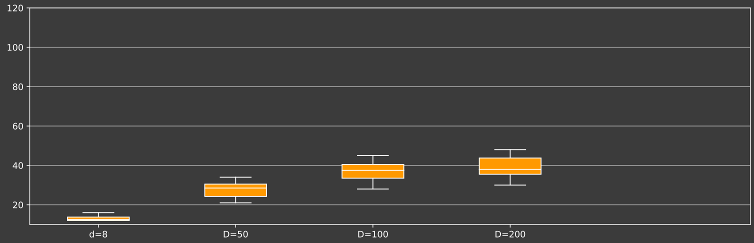


Stochos without SIEMBO:



Borehole optimization - sample comparison

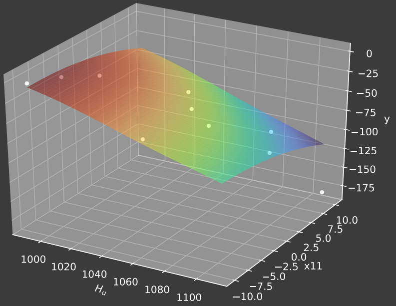
Stochos with SIEMBO:



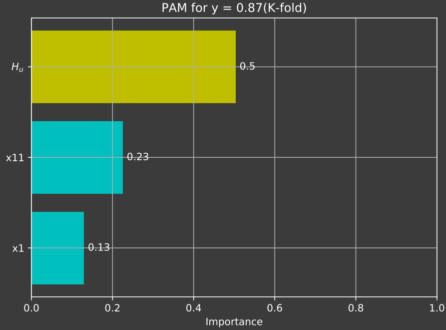
- Also the number of necessary samples can be partially reduced

Borehole optimization - $N = 11, D = 50$

Surface plot:

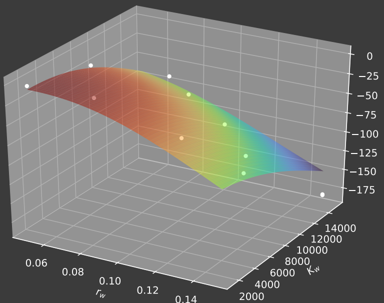


Sensitivity:

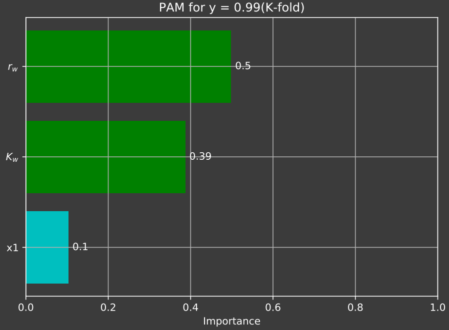


Borehole optimization - $N = 12, D = 50$

Surface plot:

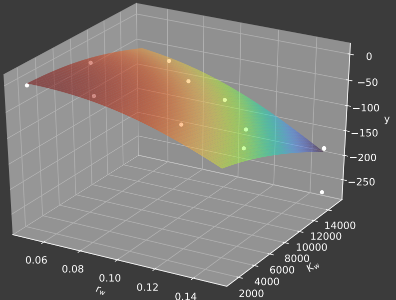


Sensitivity:

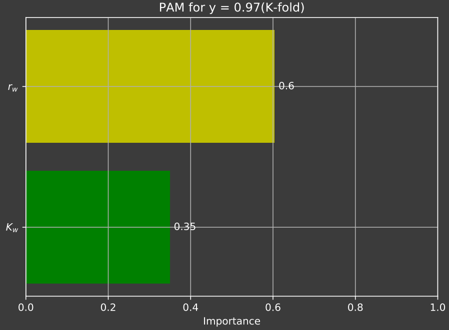


Borehole optimization - $N = 13, D = 50$

Surface plot:

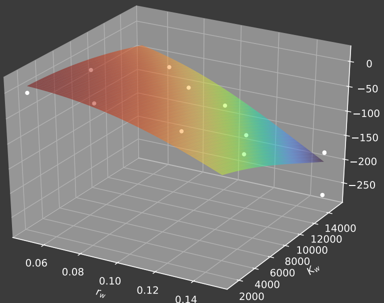


Sensitivity:

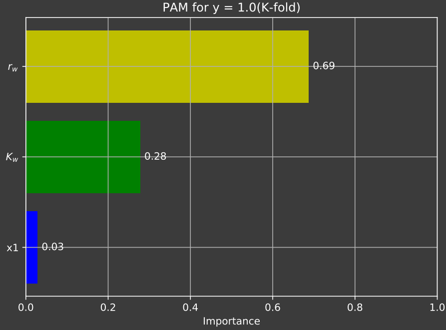


Borehole optimization - $N = 14, D = 50$

Surface plot:

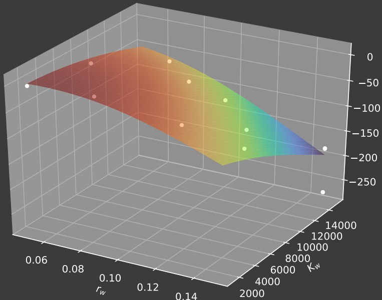


Sensitivity:

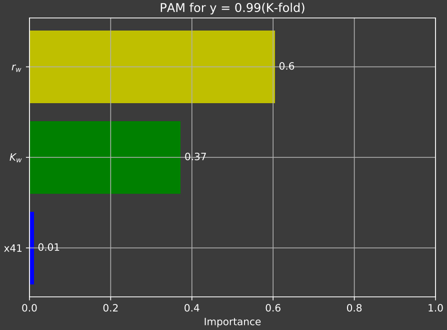


Borehole optimization - $N = 15, D = 50$

Surface plot:

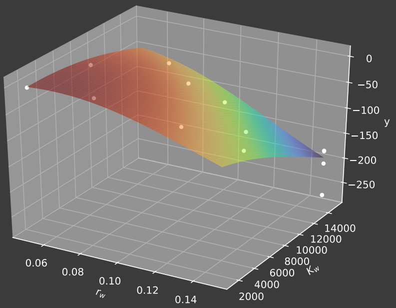


Sensitivity:

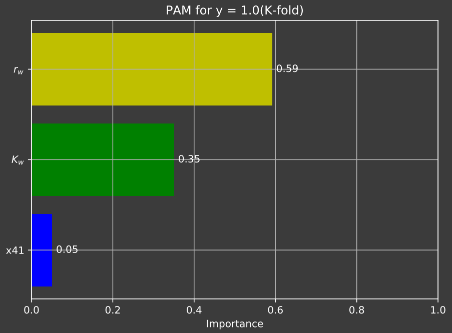


Borehole optimization - $N = 16, D = 50$

Surface plot:

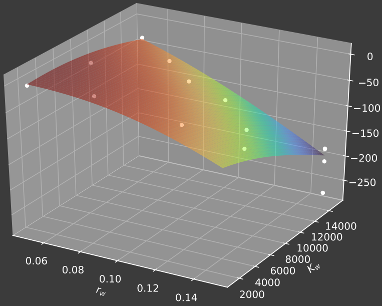


Sensitivity:

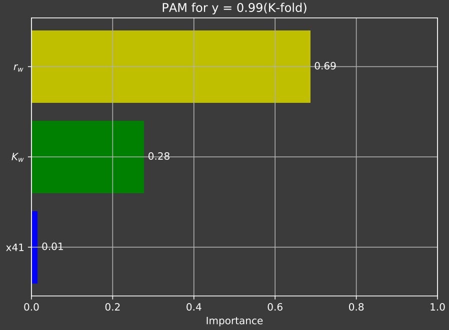


Borehole optimization - $N = 17, D = 50$

Surface plot:

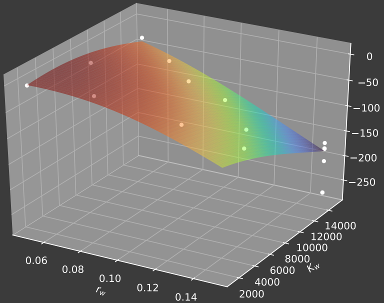


Sensitivity:

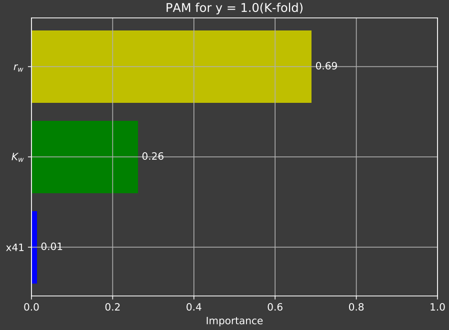


Borehole optimization - $N = 18, D = 50$

Surface plot:

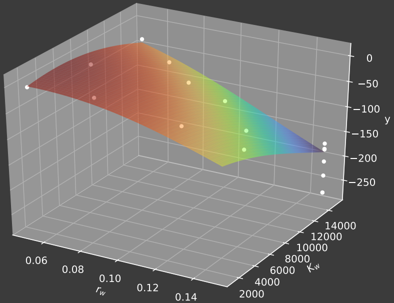


Sensitivity:

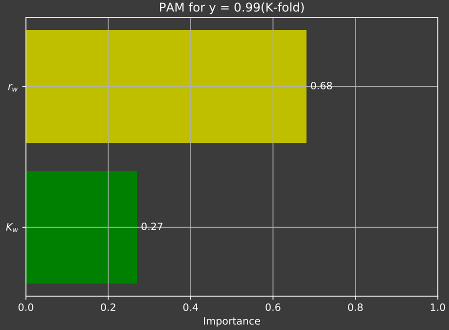


Borehole optimization - $N = 19, D = 50$

Surface plot:

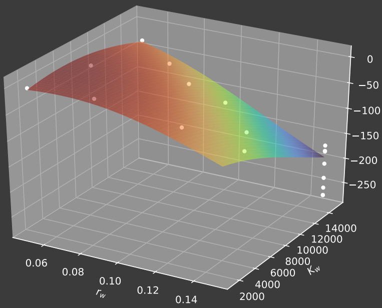


Sensitivity:

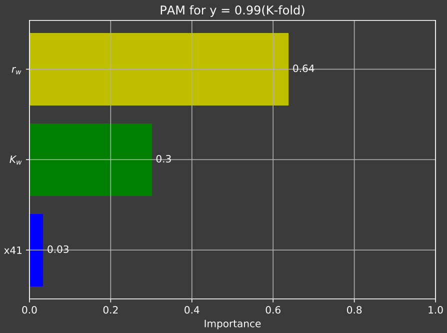


Borehole optimization - $N = 20, D = 50$

Surface plot:

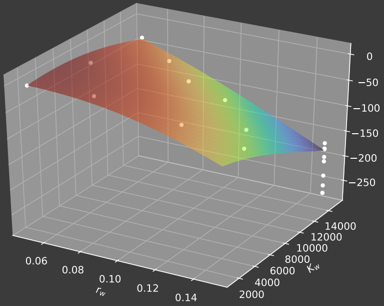


Sensitivity:

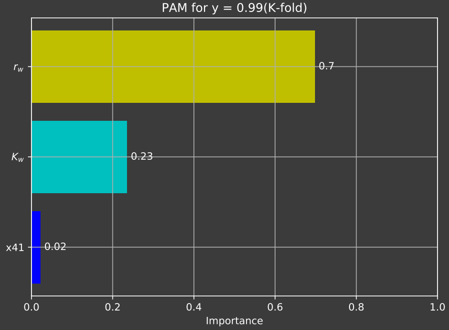


Borehole optimization - $N = 21, D = 50$

Surface plot:

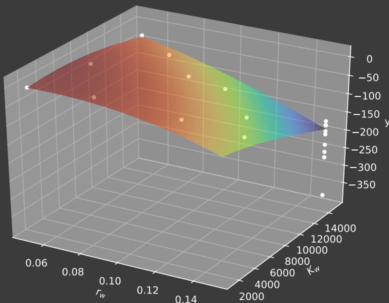


Sensitivity:

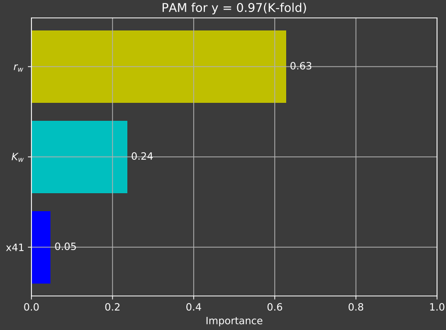


Borehole optimization - $N = 22, D = 50$

Surface plot:

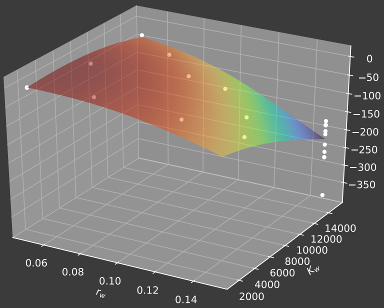


Sensitivity:

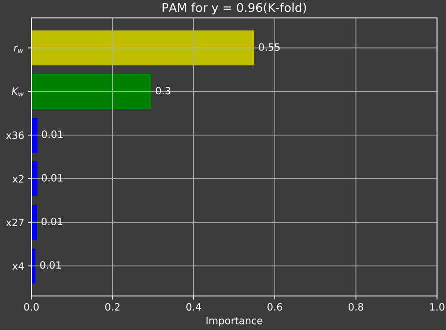


Borehole optimization - $N = 23, D = 50$

Surface plot:

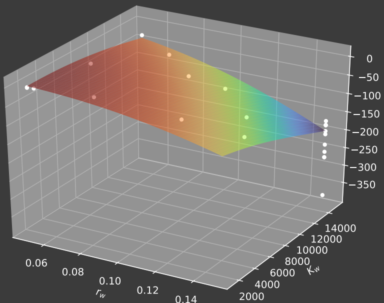


Sensitivity:

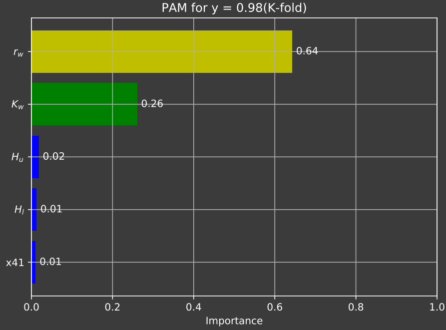


Borehole optimization - $N = 24, D = 50$

Surface plot:

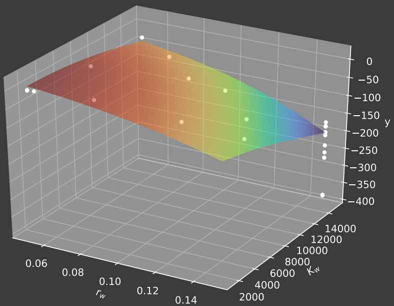


Sensitivity:

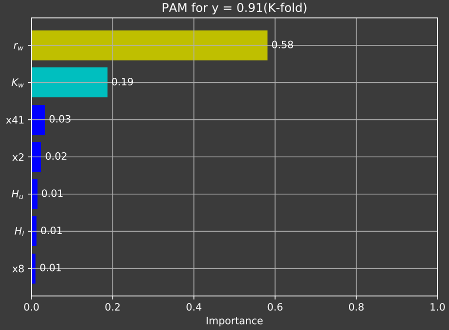


Borehole optimization - $N = 25, D = 50$

Surface plot:

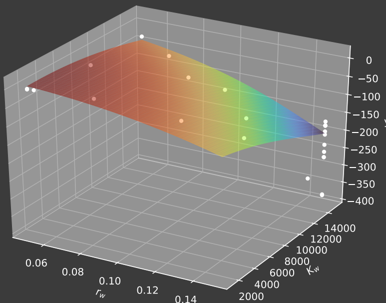


Sensitivity:

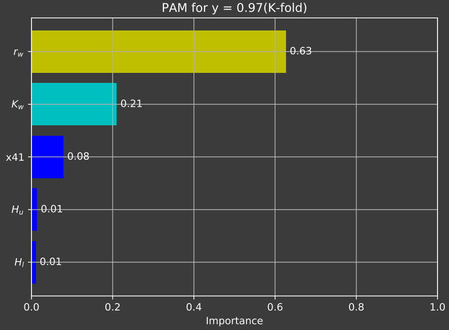


Borehole optimization - $N = 26, D = 50$

Surface plot:

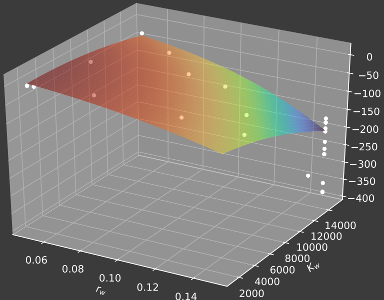


Sensitivity:

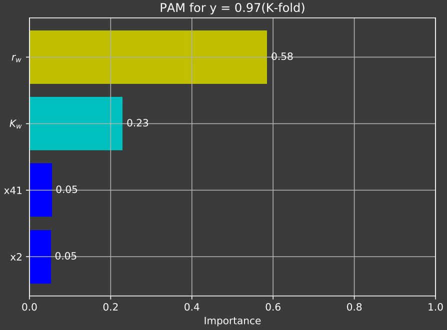


Borehole optimization - $N = 27, D = 50$

Surface plot:

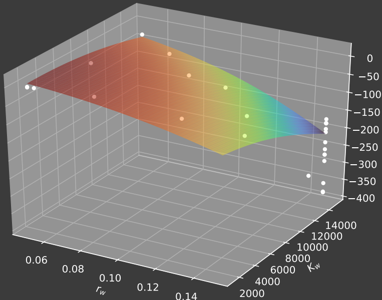


Sensitivity:

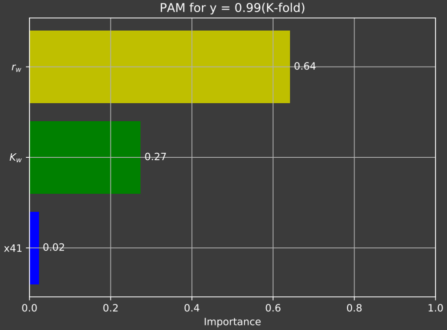


Borehole optimization - $N = 28, D = 50$

Surface plot:

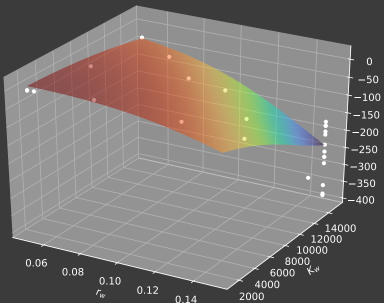


Sensitivity:

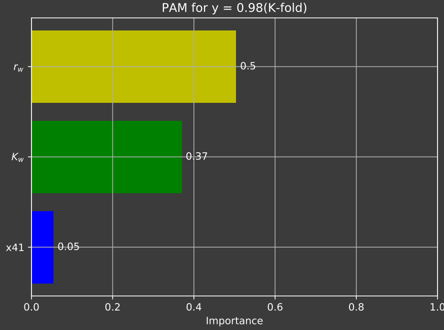


Borehole optimization - $N = 29, D = 50$

Surface plot:

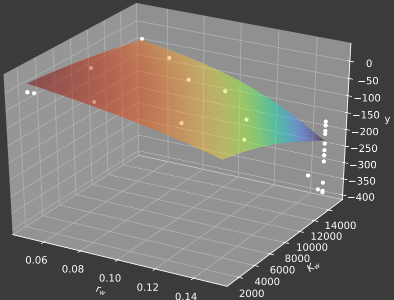


Sensitivity:

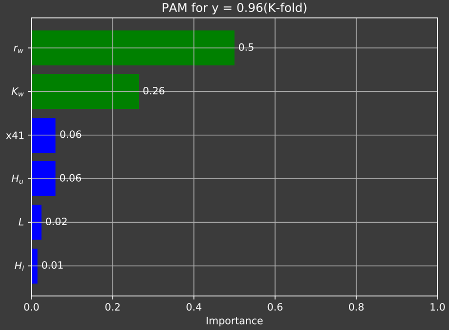


Borehole optimization - $N = 30, D = 50$

Surface plot:



Sensitivity:



Summary

- Optimization problems can be difficult to solve with increasing number of parameters
- Gradients are often not available in many practical examples
- Typically not all parameters are important or at least equally important to get a good solution
- SIEMBO tries to find the optimal embedding space in order to solve the optimization problem in the important subspace
- This allows to solve even high-dimensional optimization problems with only a few design evaluations
- It will be available within one of the next releases for Stochos inside optiSLang

Contact

Kevin Cremanns
Dirk Roos

kevin.cremanns@probaligence.com
dirk.roos@hs-niederrhein.de

PROBABILISTIC INTELLIGENCE

PI Probaligence GmbH

Hochschule Niederrhein
University of Applied Sciences

IMH
Institut für Modellbildung
und Hochleistungsrechnen
Institute of Modelling
and High-Performance Computing