Bayesian Optimization

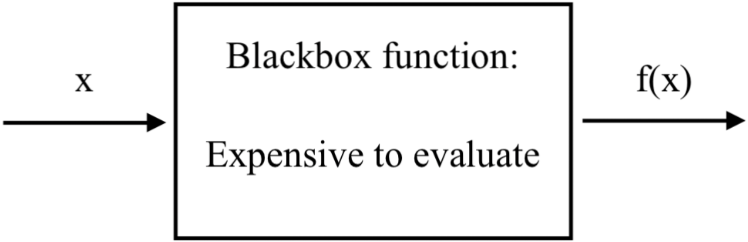
Yazid Janati El Idrissi

LIPN

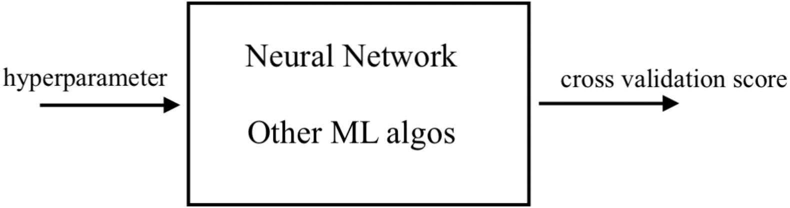
2019

Table of Contents

Context



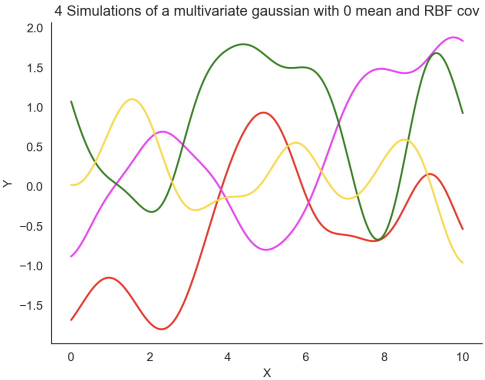
Context



hyperparameter
Neural Network
Other ML algos
cross validation score

Gaussian Process

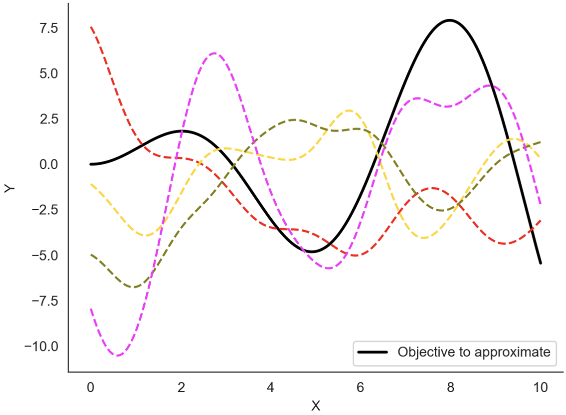
$\{\hat{f}(x_t)\}_{t\in\mathbb{R}}$ is a gaussian process, i.e. any finite collection is a gaussian vector.



$$\{\hat{f}(x_t)\}_{t\in\mathcal{I}} \sim \mathcal{N}(\mathbf{0}, \Sigma), \text{ where } \Sigma_{i,j} := \exp(-|x_i - x_j|^2/2l^2)$$

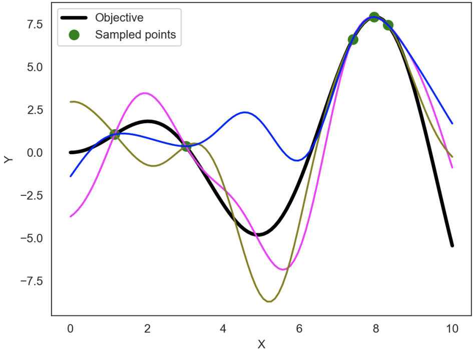
Gaussian Process

Can we approximate an objective f using gaussian processes?



Gaussian Process

We need to add **knowledge** about the objective, i.e. its values at some points.



Can be done by considering the posterior distribution

$$\hat{f}(x)|\hat{f}(x_1), \dots, \hat{f}(x_n)$$

Gaussian Process Regression: Model

We suppose that :

$$\hat{f}(x_1), ..., \hat{f}(x_n) \sim \mathcal{N}(\mu_0(x_{1:n}), \Sigma_0(x_{1:n}))$$

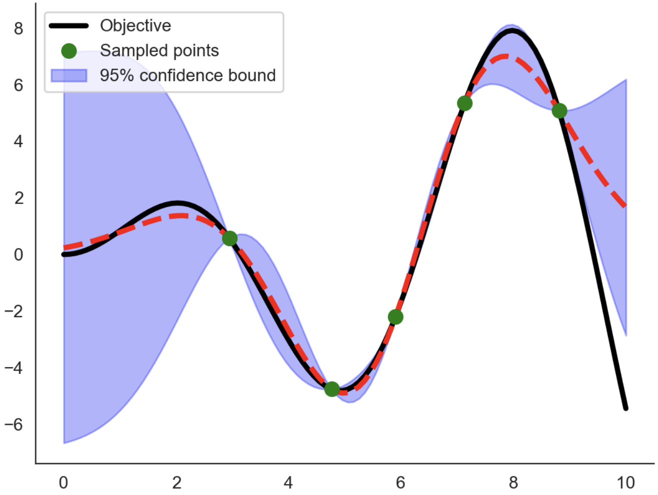
where: $\Sigma_0(x_{1:n})_{i,j} = \Sigma_0(x_i, x_j)$

We can access the conditional distribution [1] $\mathbf{\hat{f}(x)}|\mathbf{\hat{f}(x_{1:n})}$, $\forall x \in \mathcal{D}$:

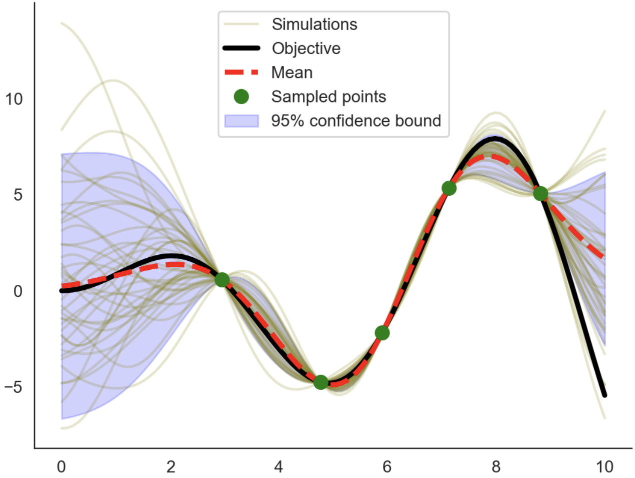
$$\hat{f}(x)|\hat{f}(x_{1:n}) \sim \mathcal{N}(\mu_n(x), \sigma_n(x)^2)$$

Gaussian Process Regression: Model

Prediction : mean and variance at each x



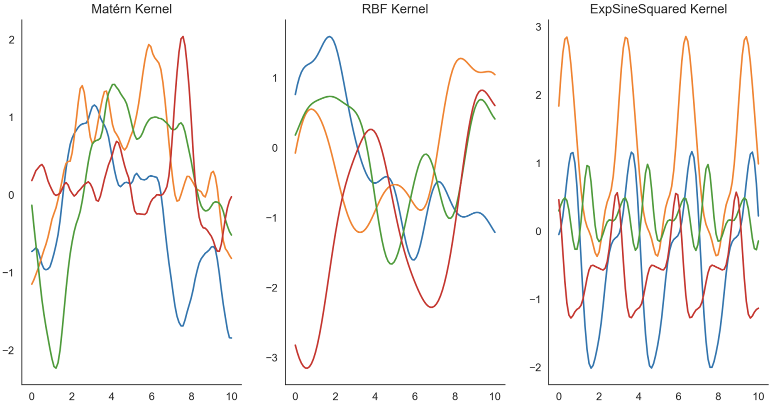
Gaussian Process Regression: Model



Gaussian Process Regression: Model selection

Hyperparameters?

Which kernel to use: RBF, Matèrn, Exp Spine?

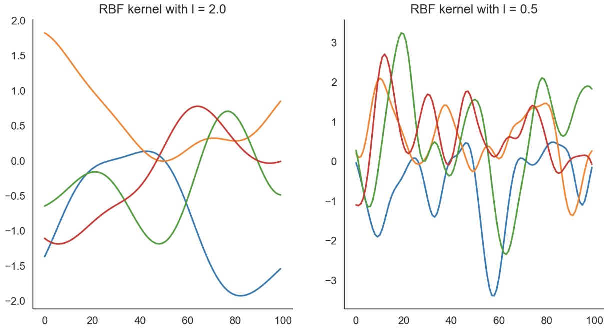


Gaussian Process Regression: Model selection

Hyperparameters?

How to choose the hyperparameters of the kernel?

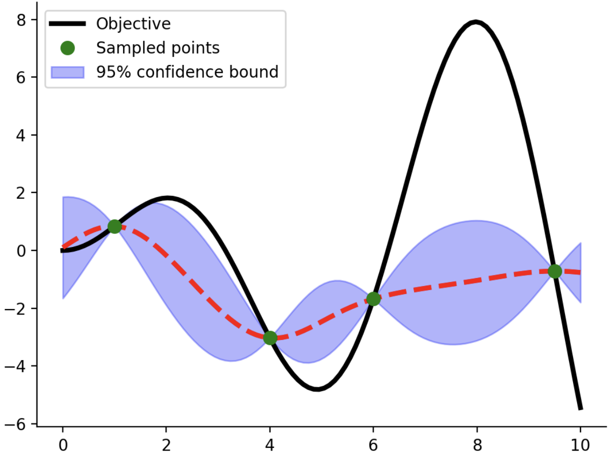
RBF kernel: $cov(x_i, x_j) := \exp(-|x_i - x_j|^2/2l^2)$



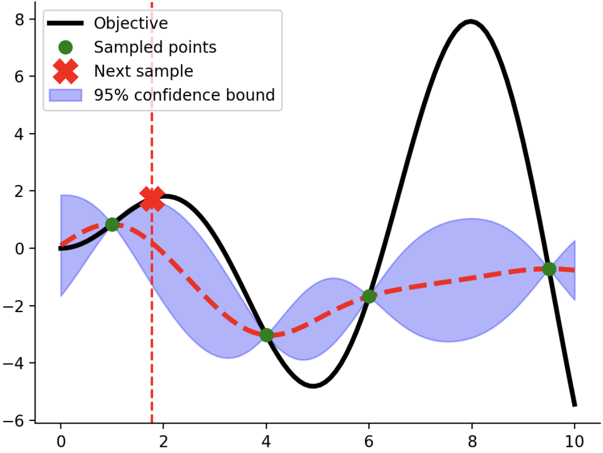
via likelihood maximization [1]:

$$\log p(y|X, \theta) = \frac{1}{2} y^T K_y^{-1} y - \frac{1}{2} \log |K_y| - \frac{n}{2} \log(2\pi)$$

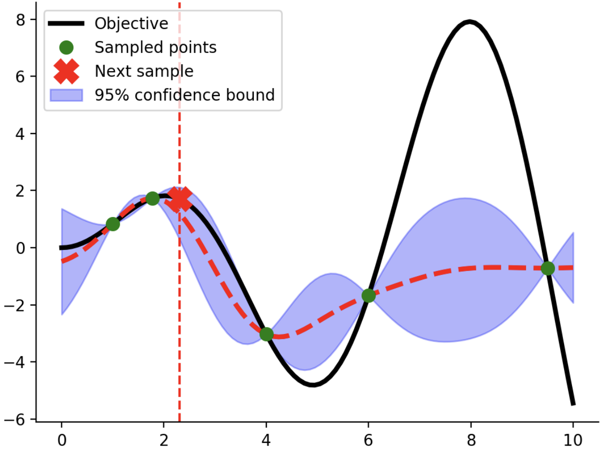
Bayesian Optimization: Walkthrough



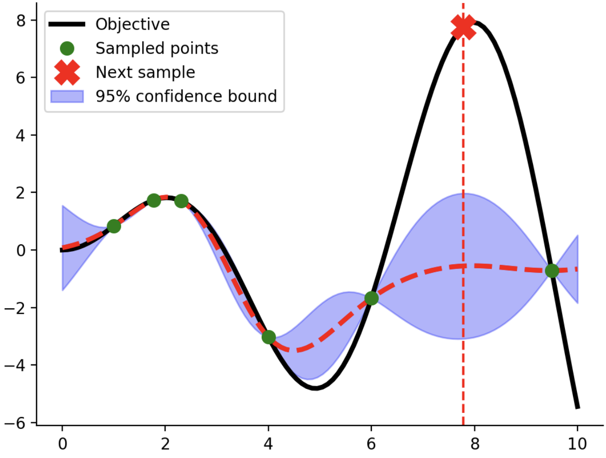
Bayesian Optimization: Walkthrough



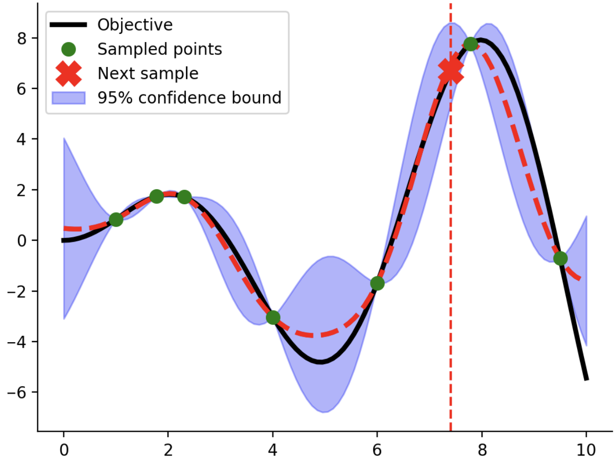
Bayesian Optimization: Walkthrough



Bayesian Optimization: Walkthrough



Bayesian Optimization: Walkthrough



Bayesian Optimization: Summary

Instead of directly optimizing the objective, we instead optimize a **surrogate**: e.g. **Upper Confidence Bound** [3]:

$$\mathrm{UCB}_n(x) = \mu_n(x) + \beta_n^* \sigma_n(x)$$

Algorithm [2]:

- Fit a **gaussian process regression** model using the known points $A = \{(x_i, f(x_i))\}_i$.
- Solve $\operatorname{argmax}_{x \in \mathcal{D}} \mu_n(x) + \beta_n \sigma_n(x)$ and append solution to A.
- Restart.

Return arg of the best value visited.

*typically [3]: $\beta_n = \sqrt{\log(n)}$.

Bayesian Optimization: Applications

Simple regret : $|f(x^*) - \text{argmax}_{x \in \mathcal{D}} f(x)|$

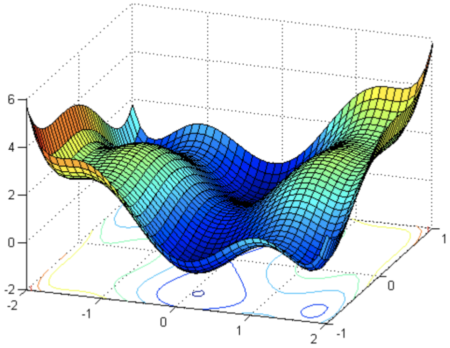
Experiment: run Bayesian Optimization 10 times and average the results.

- Two synthetic functions: Ackley and Hartmann
- One real world example: DBSCAN on car scenarios (Renault's Data).

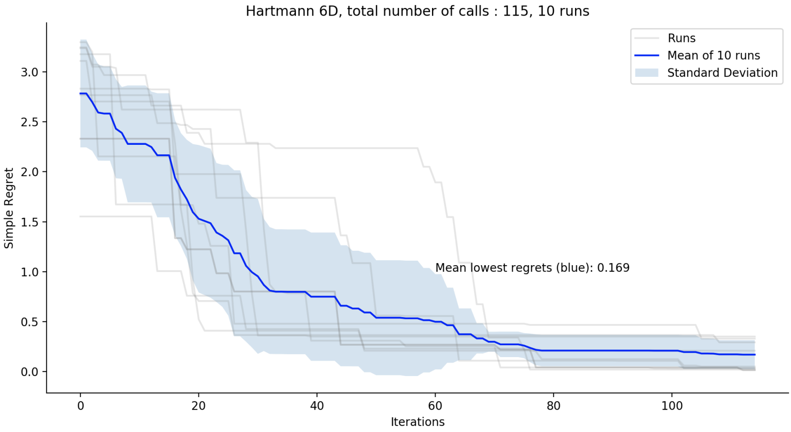
Bayesian Optimization: Applications

Example: Hartmann function:

- Domain : $[0, 1]^6$
- Lowest regret on 10 runs: 0.01



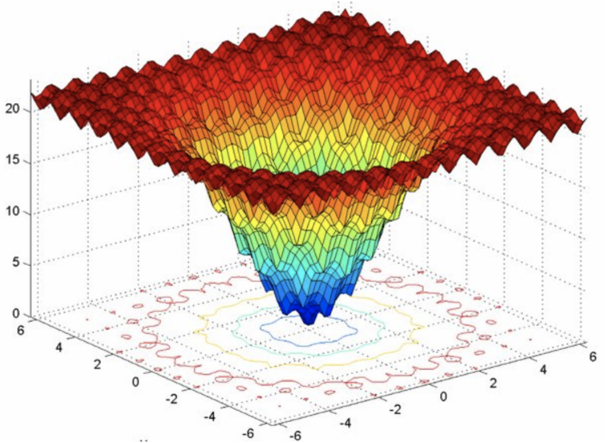
Bayesian Optimization: Applications



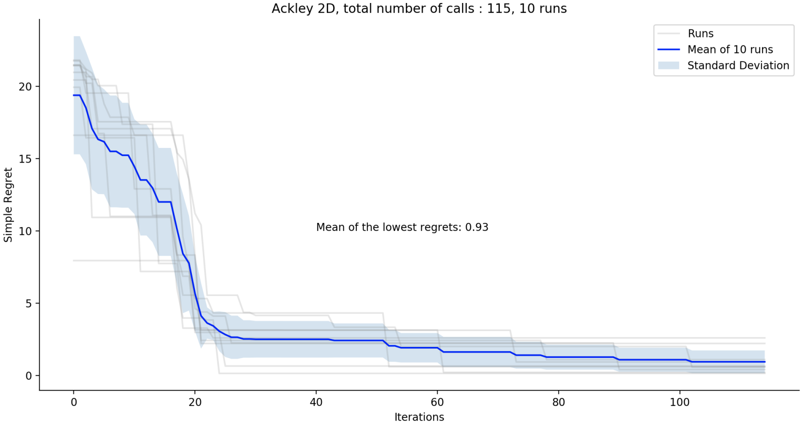
Bayesian Optimization: Applications

Example: Ackley function:

- Domain : $[-32, 32]^2$
- Lowest regret on 10 runs: 0.13



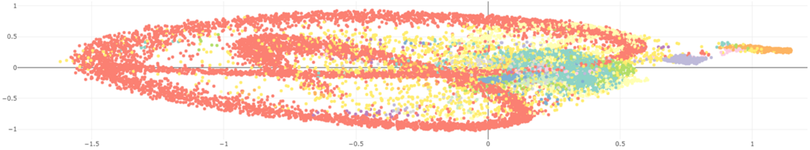
Bayesian Optimization: Applications



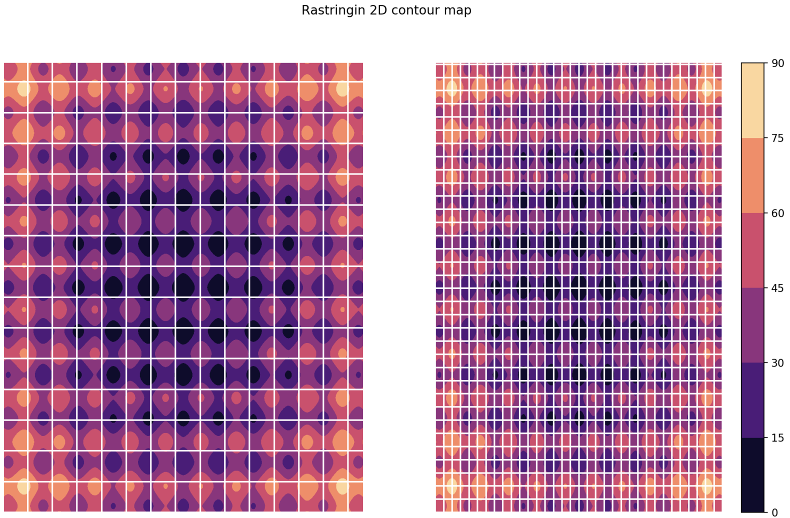
Bayesian Optimization: Applications

Example:

- Clustering of scenarios after performing PCA or MDS.
- Optimization with respect to *eps* and *num samples*
- Space: $[0,1]^2 \times [0,200]$

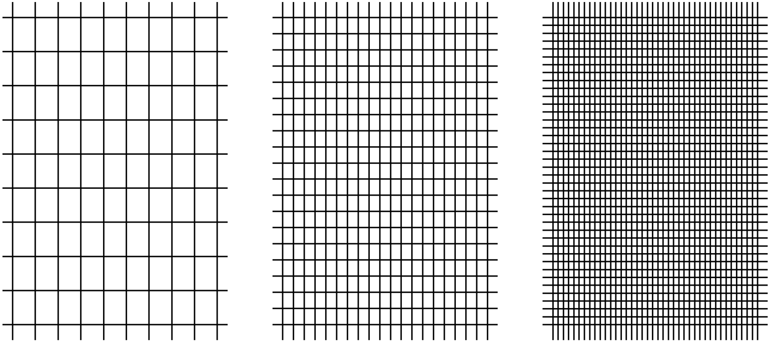


Bayesian Optimization: A novel approach



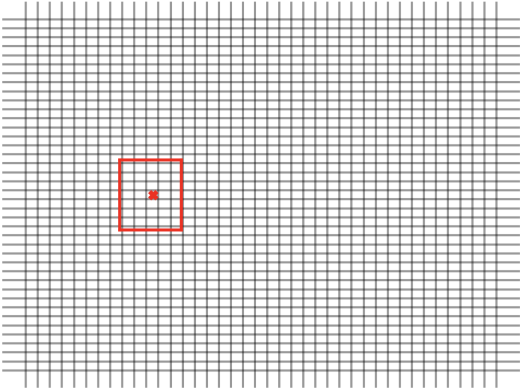
Bayesian Optimization: A novel approach

- Optimize on grids with decreasing step size. (e.g. divide step size by two at each step)

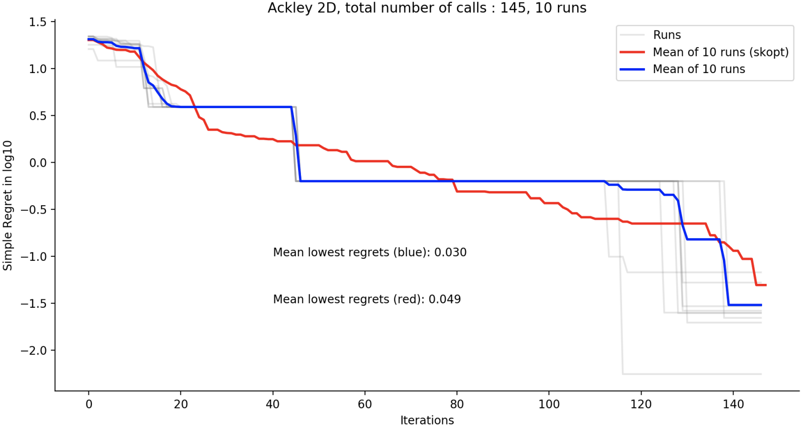


Bayesian Optimization: A novel approach

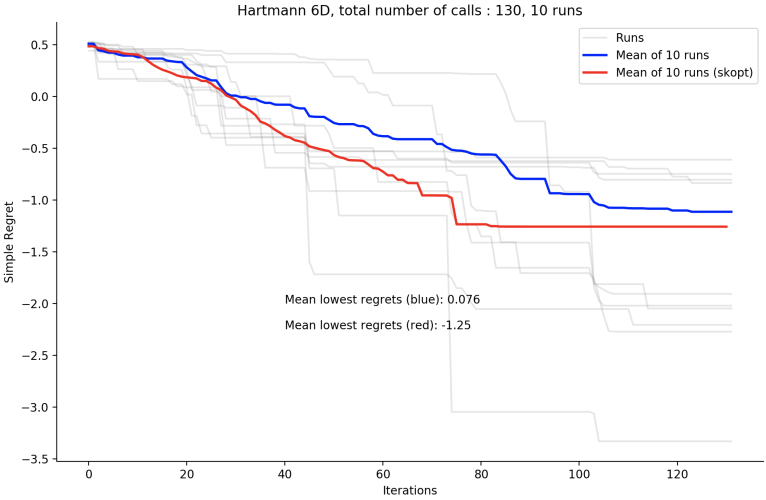
- Select best point found so far and optimize **continuously** in a ball centered around it.



Bayesian Optimization: A novel approach



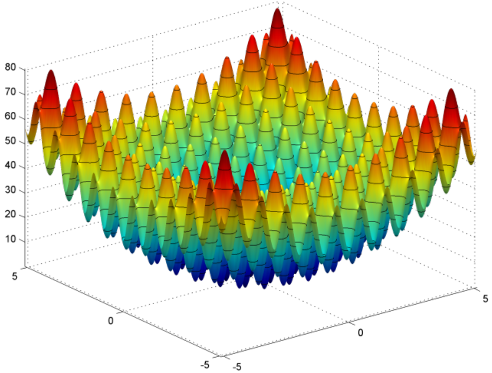
Bayesian Optimization: A novel approach



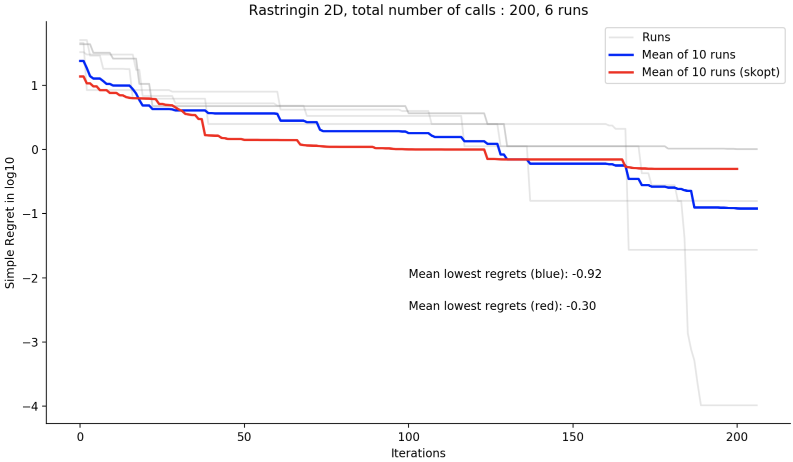
Bayesian Optimization: A novel approach

Example: Rastringin function.

- Domain : $[-5.12, 5.12]^2$
- Lowest regret on 6 runs: 1e-3



Bayesian Optimization: A novel approach



Software

Bayesian Optimization library in Scala and Python. Features:

- Discrete or continuous optimization. Mixed space also.
- Custom acquisition functions.
- Custom optimizers.
- Easy to use.

Python version available at:

`https://github.com/yazidjanati/bayestuner`

References

- Rasmussen & Williams, Gaussian Processes For Machine Learning, 2006 MIT, 2006
- Peter I. Frazier, A Tutorial On Bayesian Optimization, 2018
- Niranjan Srinivas et al., Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design, 2009