Guest Lecture for the Automatic Machine Learning Course

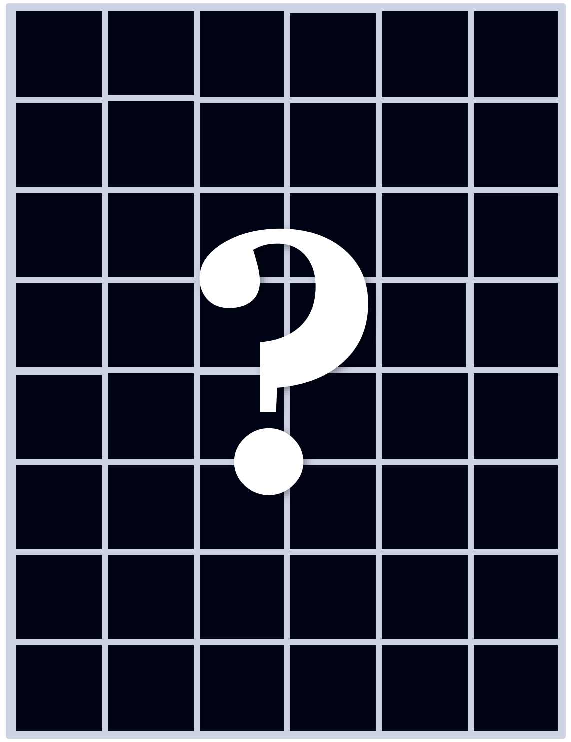
Bayesian Optimization

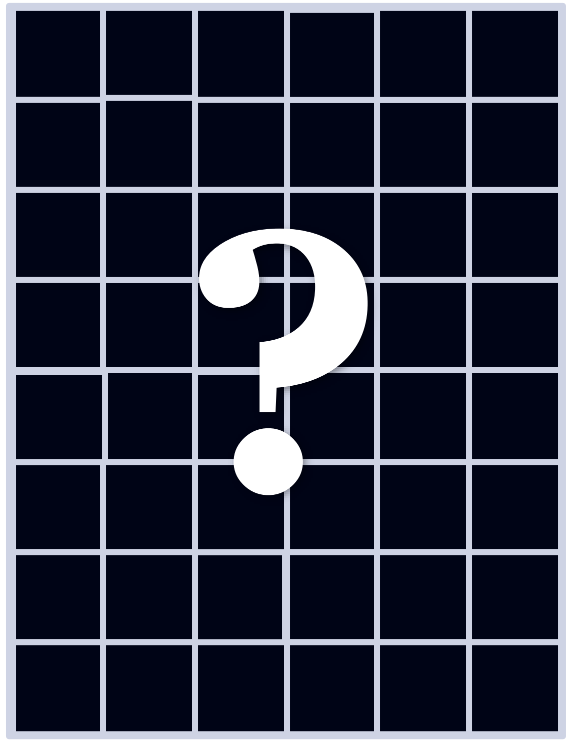
Elena Raponi

Universiteit Leiden
The Netherlands

Model-based optimization– Visual motivation

Model-based
Algorithm 1



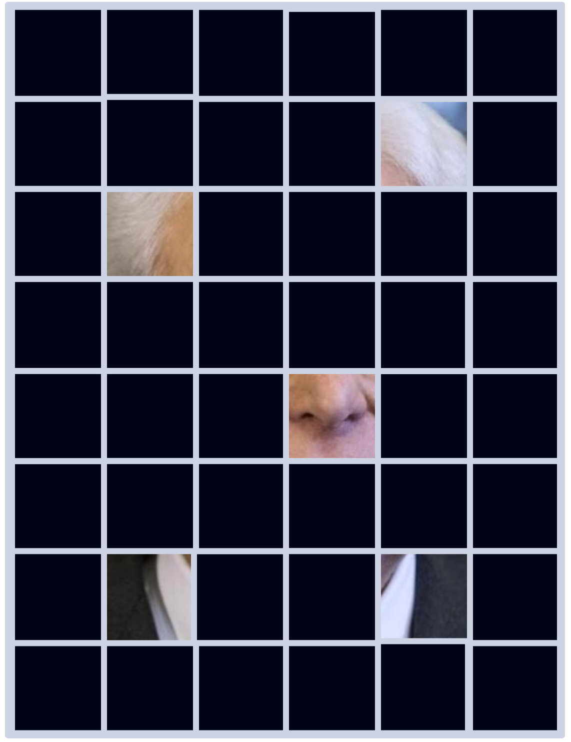


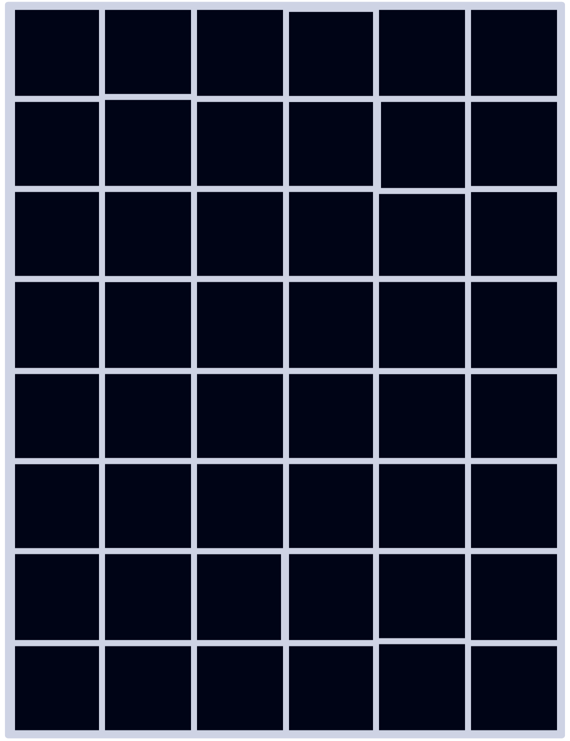
Direct search
Algorithm 2

Model-based optimization – Visual motivation

Model-based
Algorithm 1

€ 5000



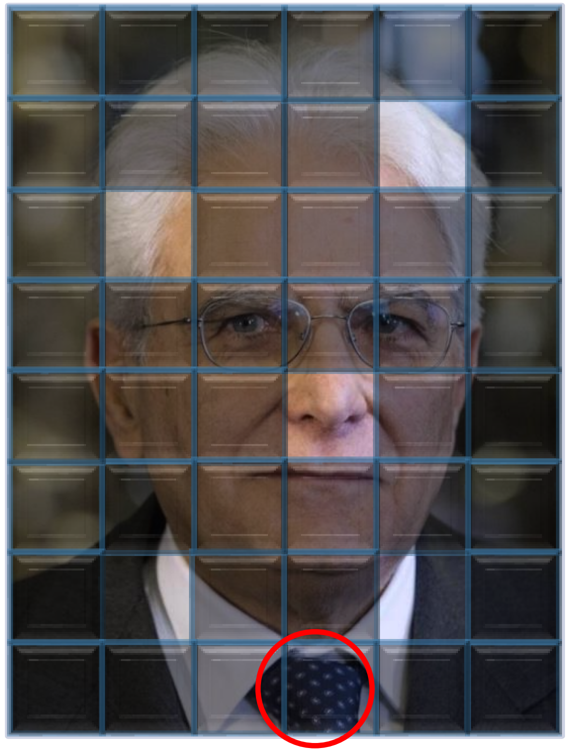


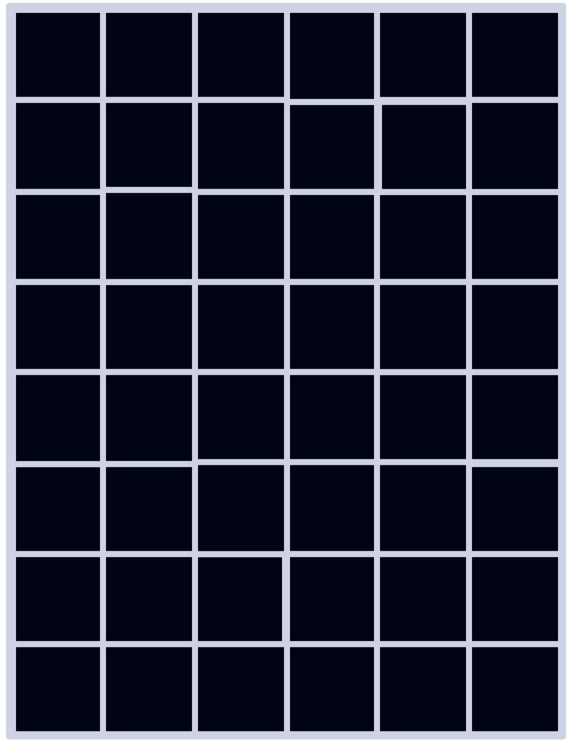
Direct search
Algorithm 2

Model-based optimization – Visual motivation

Model-based
Algorithm 1

€ 5000



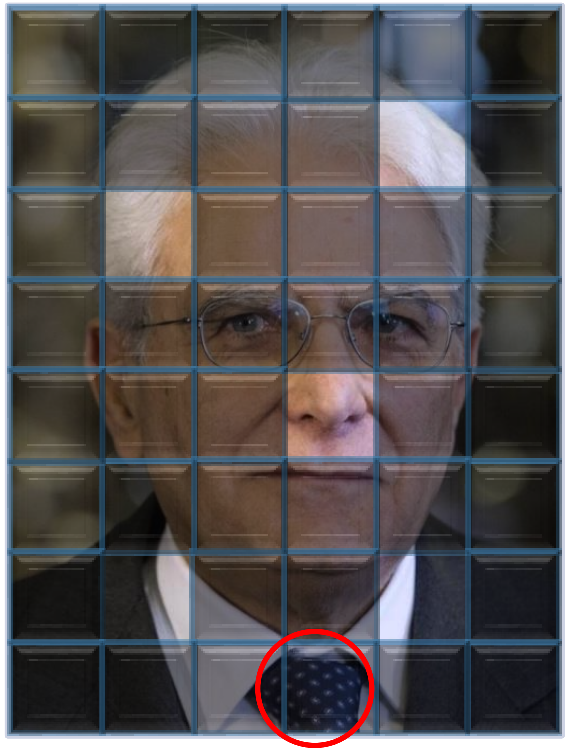


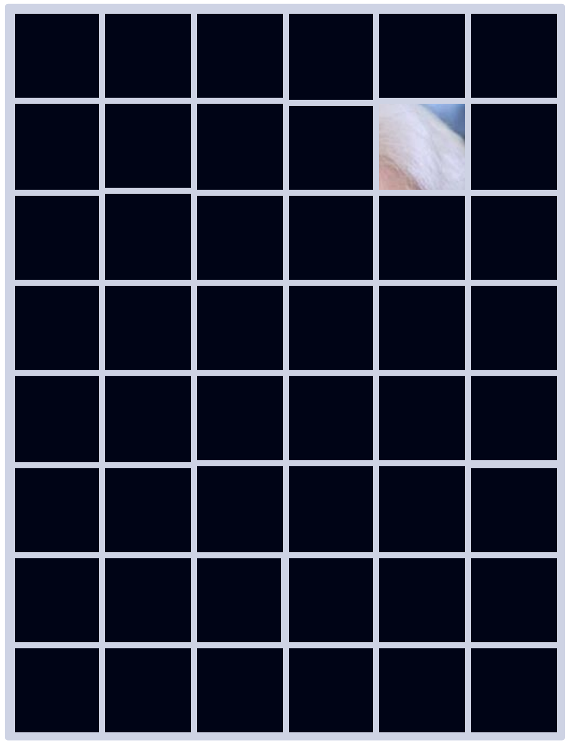
Direct search
Algorithm 2

Model-based optimization – Visual motivation

Model-based
Algorithm 1

€ 5000



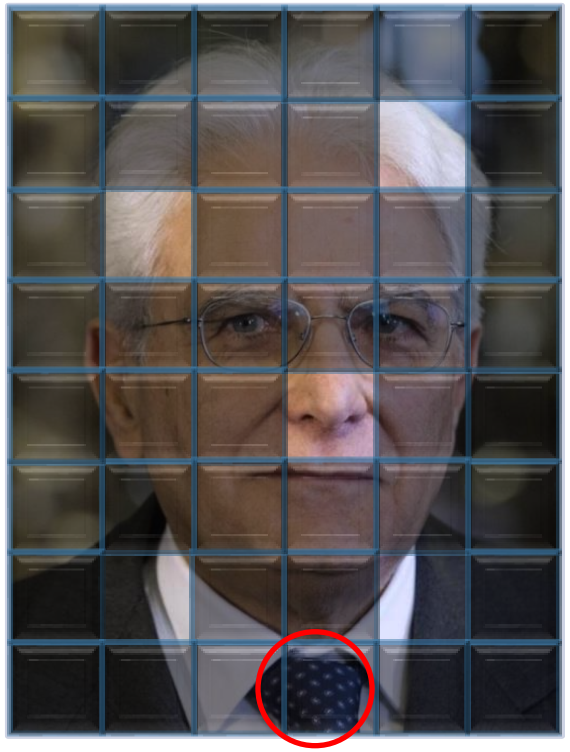


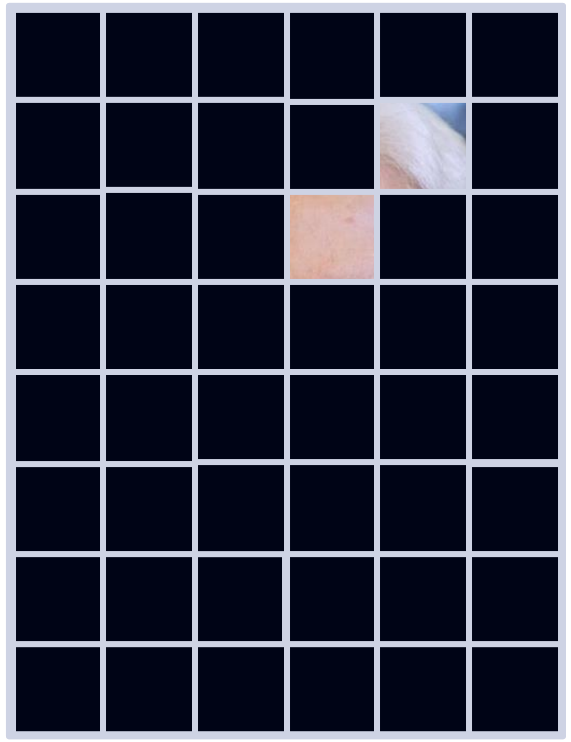
Direct search
Algorithm 2

Model-based optimization – Visual motivation

Model-based
Algorithm 1

€ 5000



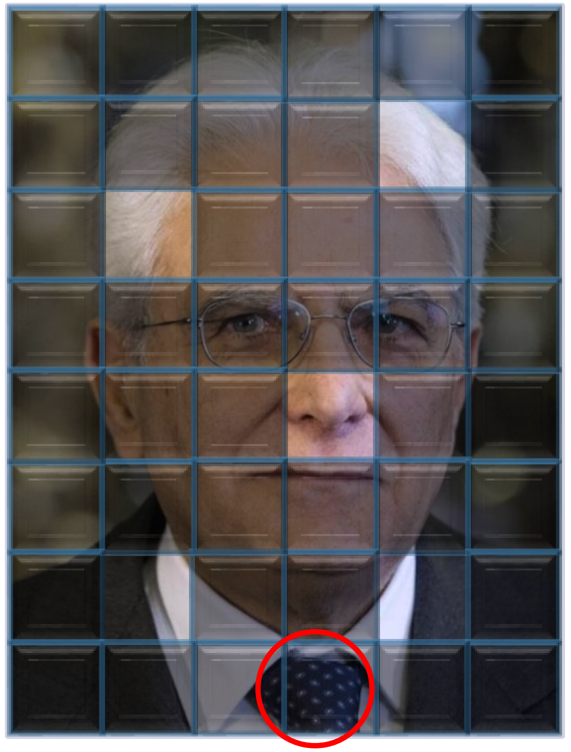


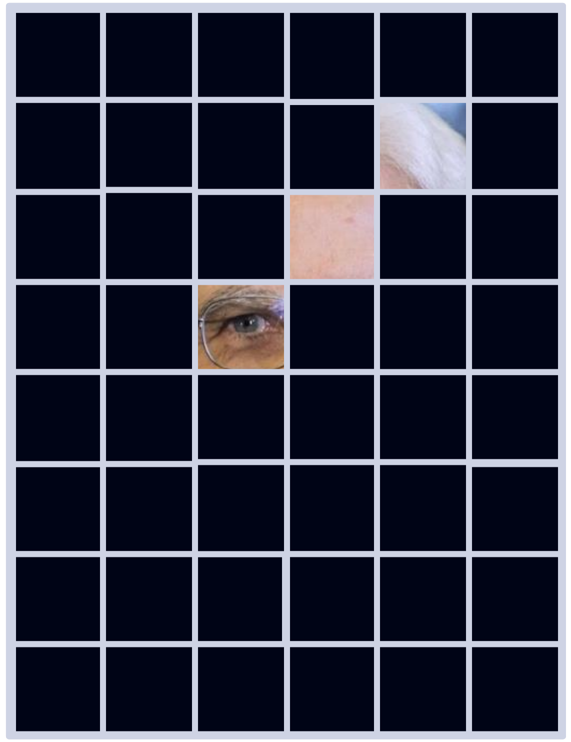
Direct search
Algorithm 2

Model-based optimization – Visual motivation

Model-based
Algorithm 1

€ 5000



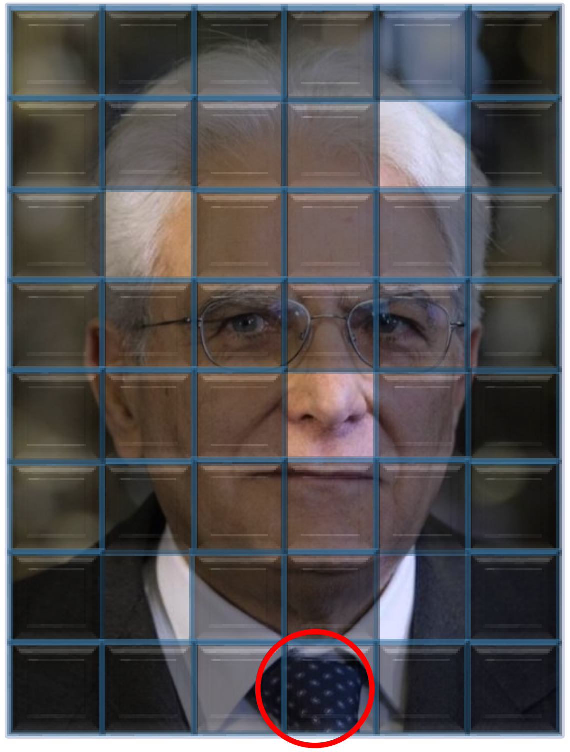


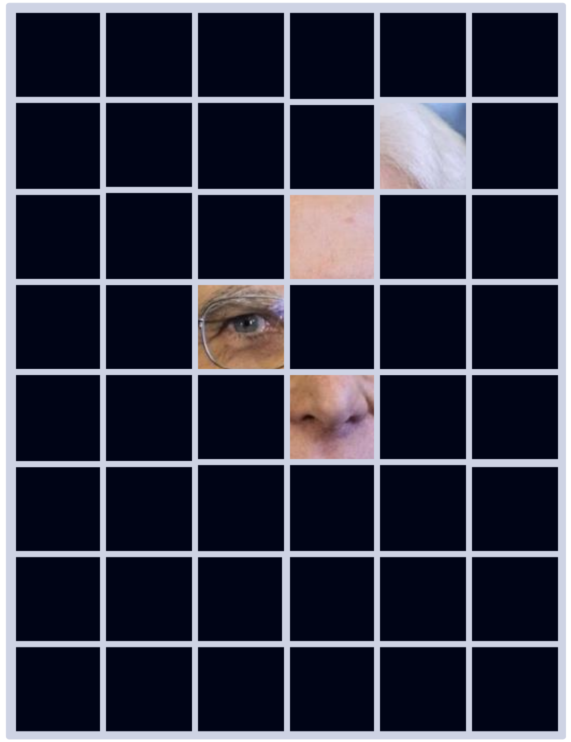
Direct search
Algorithm 2

Model-based optimization – Visual motivation

Model-based
Algorithm 1

€ 5000



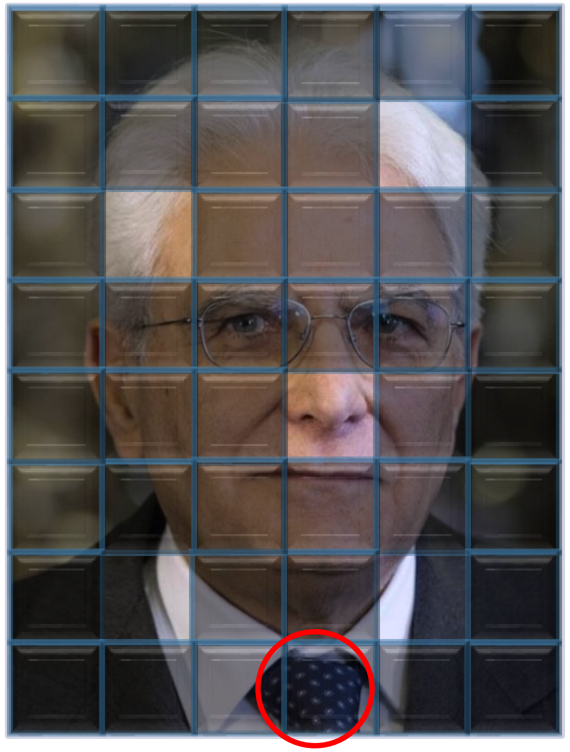


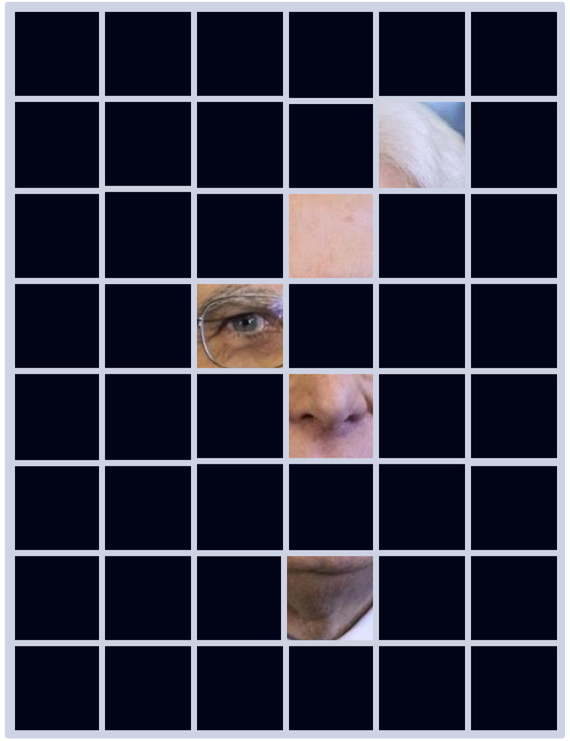
Direct search
Algorithm 2

Model-based optimization – Visual motivation

Model-based
Algorithm 1

€ 5000



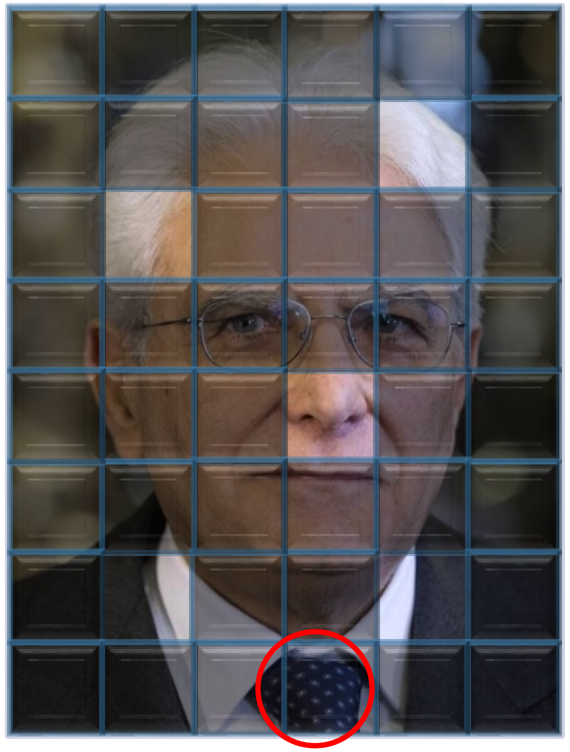


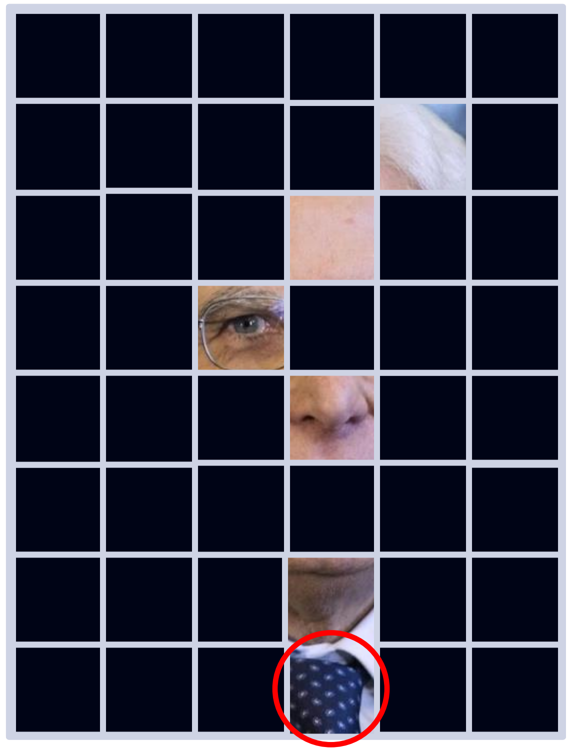
Direct search
Algorithm 2

Model-based optimization – Visual motivation

Model-based
Algorithm 1

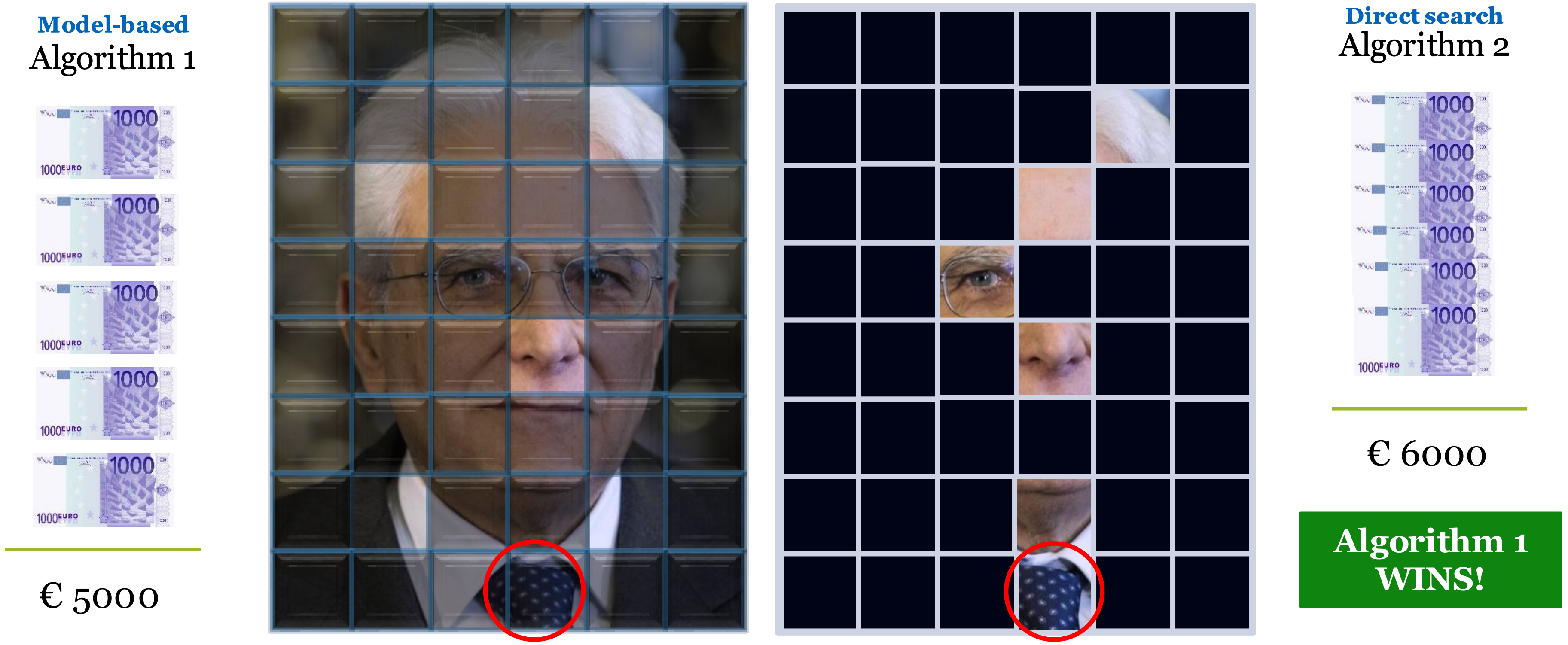
€ 5000



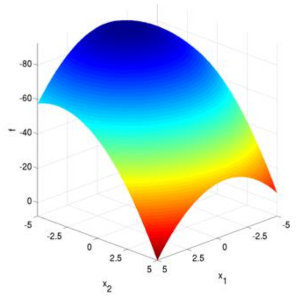


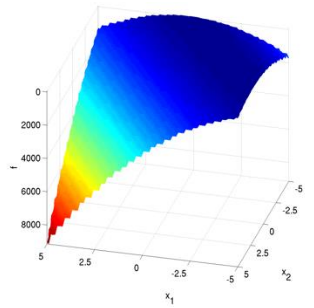
Direct search
Algorithm 2

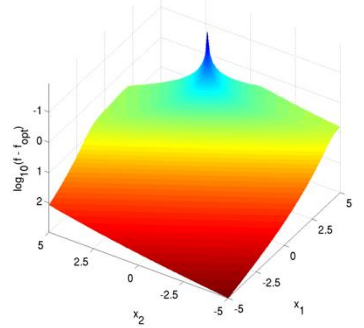
Model-based optimization – Visual motivation

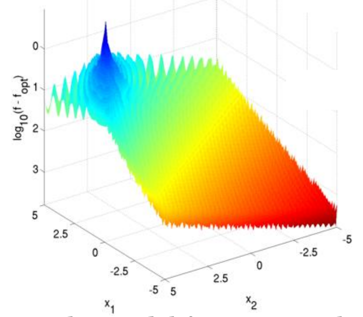


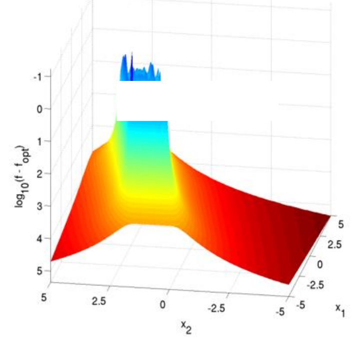
Model-based optimization – Visual motivation

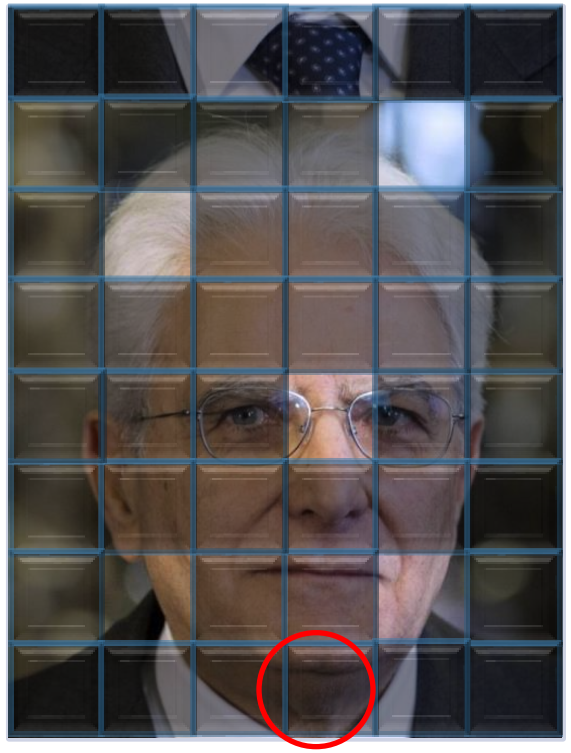

separable functions

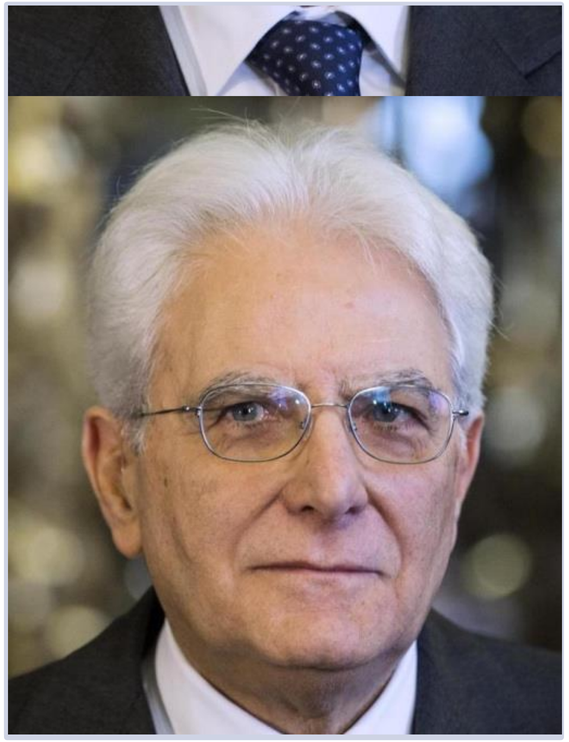

functions with low or moderate conditioning


functions with high conditioning and unimodal structure


multi-modal functions with adequate global structure


multi-modal functions with weak global structure





Remember that not all problems (function landscapes) are friendly!

The Optimization Challenge

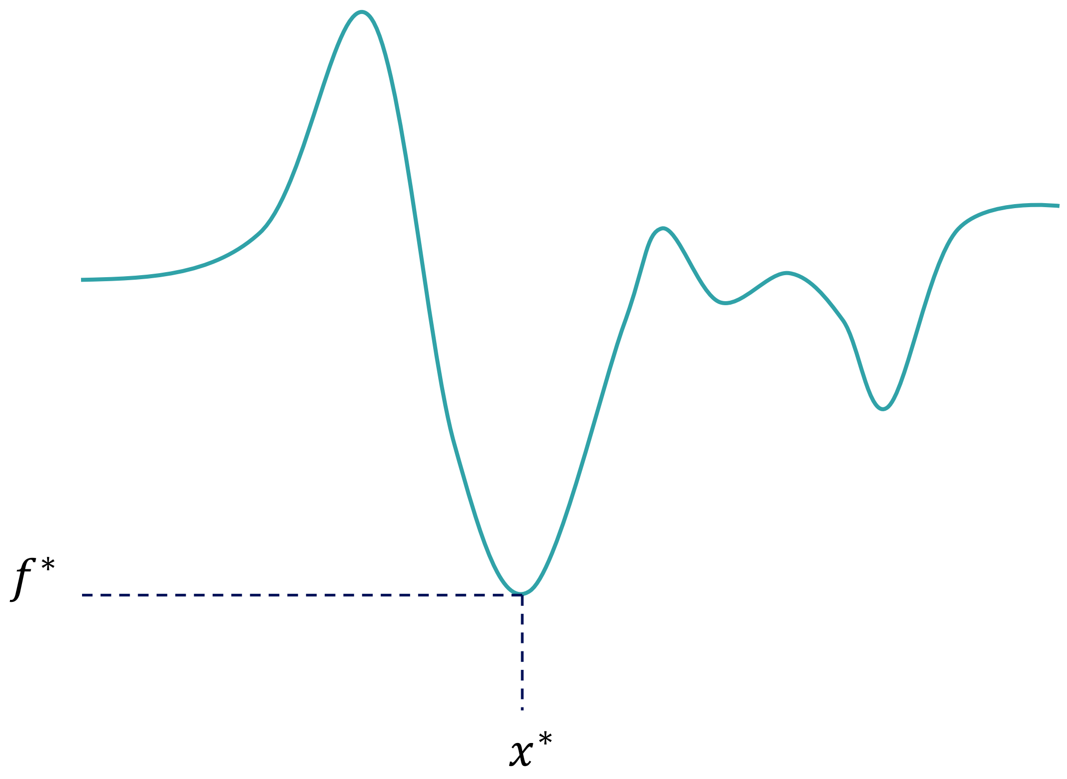
In the context of Bayesian optimization, Gaussian Processes (GPs) are used to model a real-valued function that we want to optimize.

Let $f: X \rightarrow \mathbb{R}$ be a real-valued function defined on a domain X (let's assume $X \subset \mathbb{R}^d$ for now)

We want to find a point minimizing f:

$$x^* = \underset{x \in X}{\operatorname{argmin}} f(x);$$

$$f^* = \min_{x \in X} f(x) = f(x^*)$$

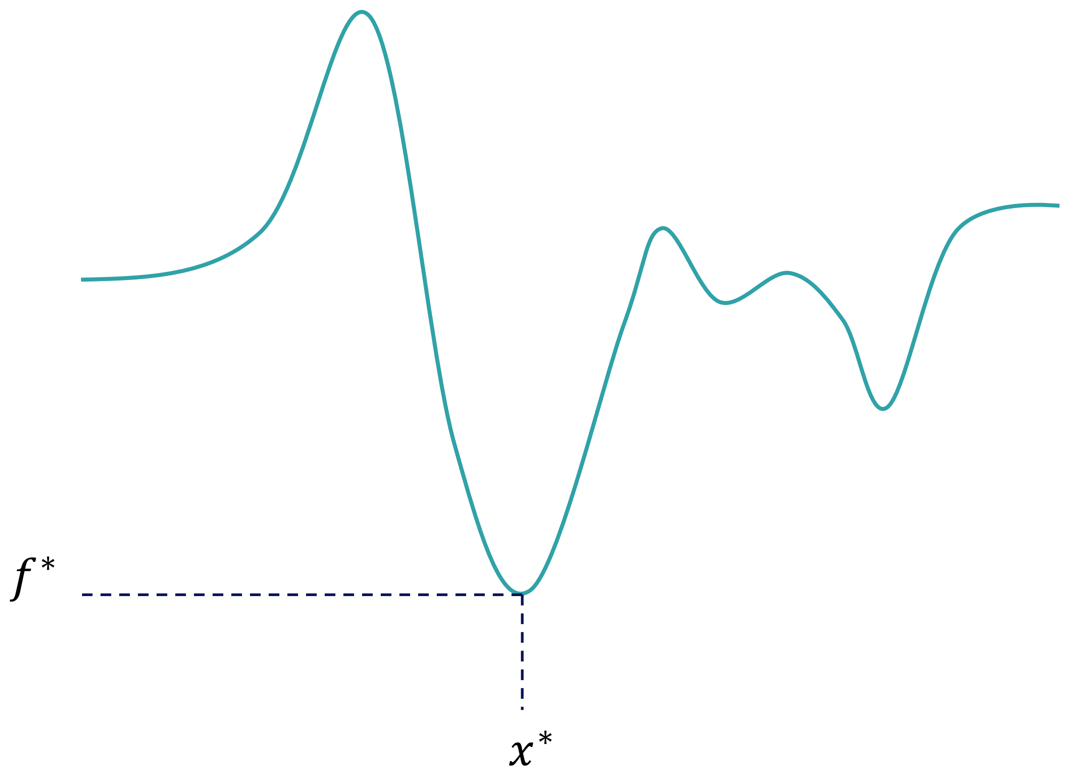


Note: we do not require that the objective function have a known functional form

The Optimization Challenge

Let $f: X \to \mathbb{R}$ be a real-valued function defined on a domain X (let's assume $X \subset \mathbb{R}^d$ for now)

- Vanilla BO usually performs well for d<20
- f's feasible set is simple, e.g., box constraints
- f is continuous but lacks special structure , e.g., concavity, that would make it easy to optimize
- f is derivative-free: evaluations do not give gradient information
- f is expensive to evaluate: the # of times we can evaluate it is severely limited
- f may be noisy. If noise is present, we'll assume it is independent and normally distributed, with common but unknown variance



Standard logical flow of BO

Target: Find $\boldsymbol{x}^* = \underset{\boldsymbol{x} \in X}{\operatorname{argmin}} f(\boldsymbol{x})$

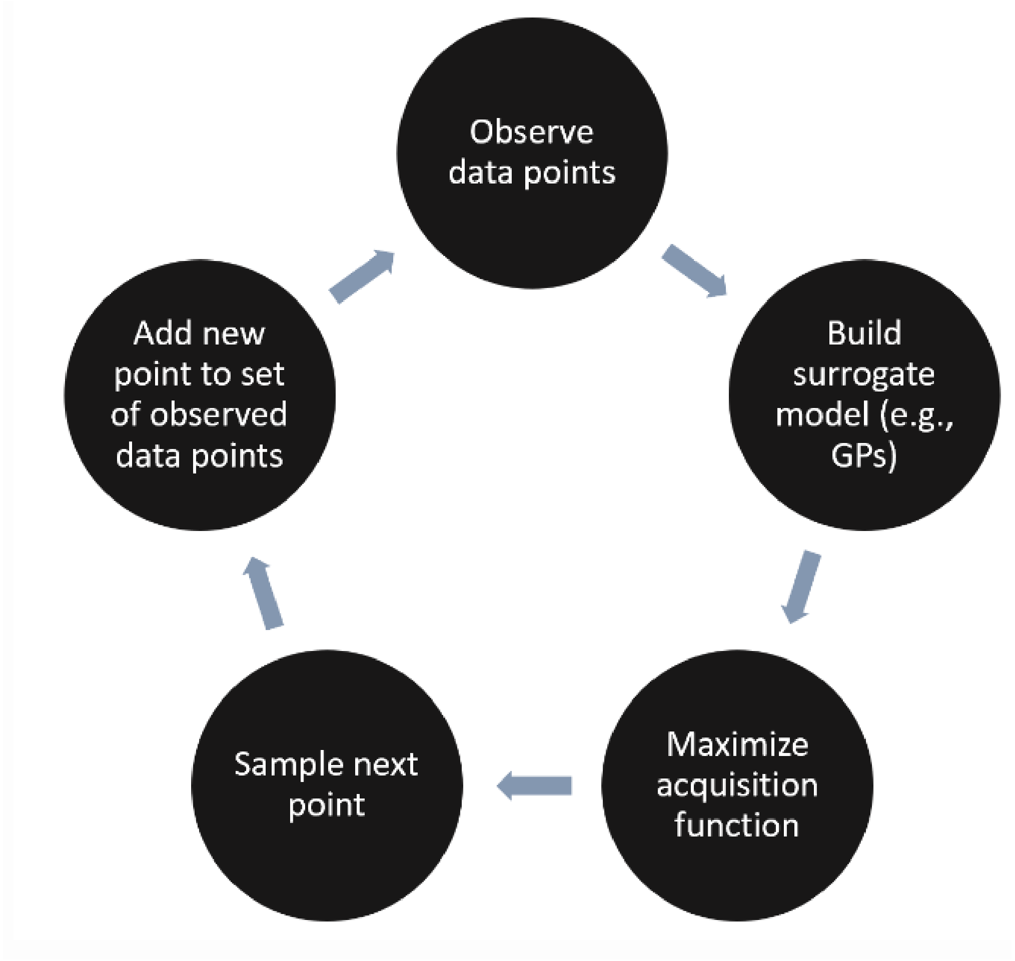
Assume a Bayesian **prior** on f (usually a Gaussian process prior)

Build the **posterior** distribution

while (budget is not exhausted) {

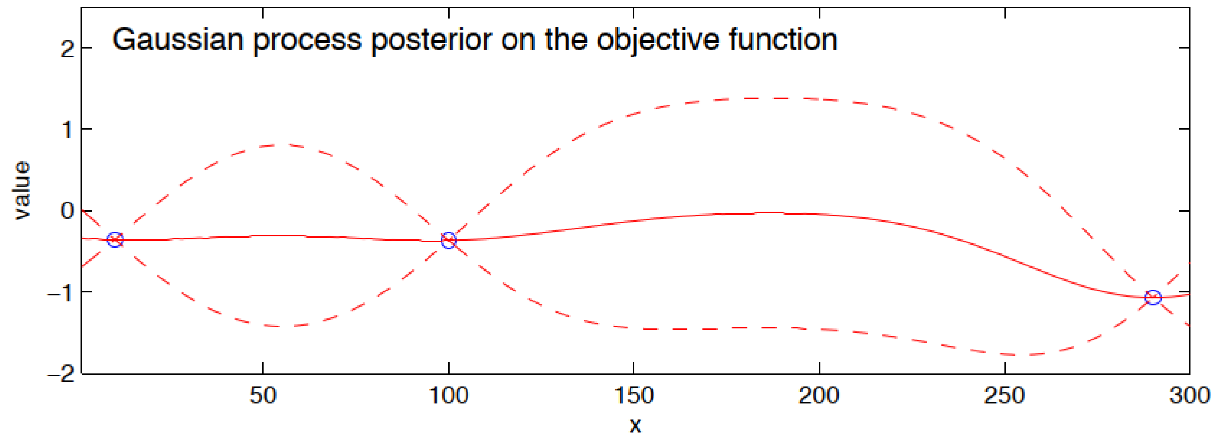
- Find x that maximizes **acquisition**(x,**posterior**)
- **Sample** x & **observe** f(x)
- **Update** the **posterior** distribution on f

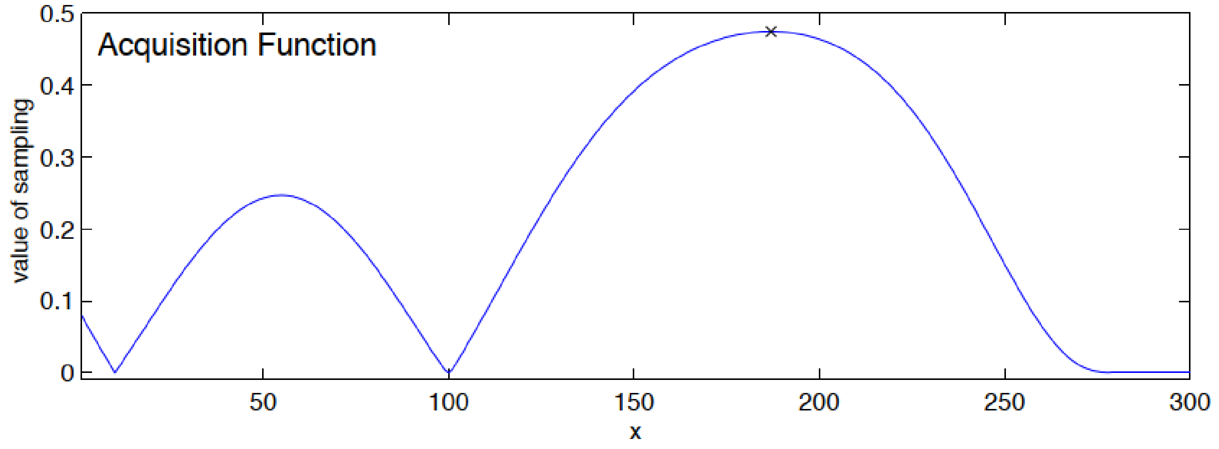
}


Observe data points
Build surrogate model (e.g., GPs)
Maximize acquisition function
Sample next point
Add new point to set of observed data points

Optimizing a 1-dimensional problem

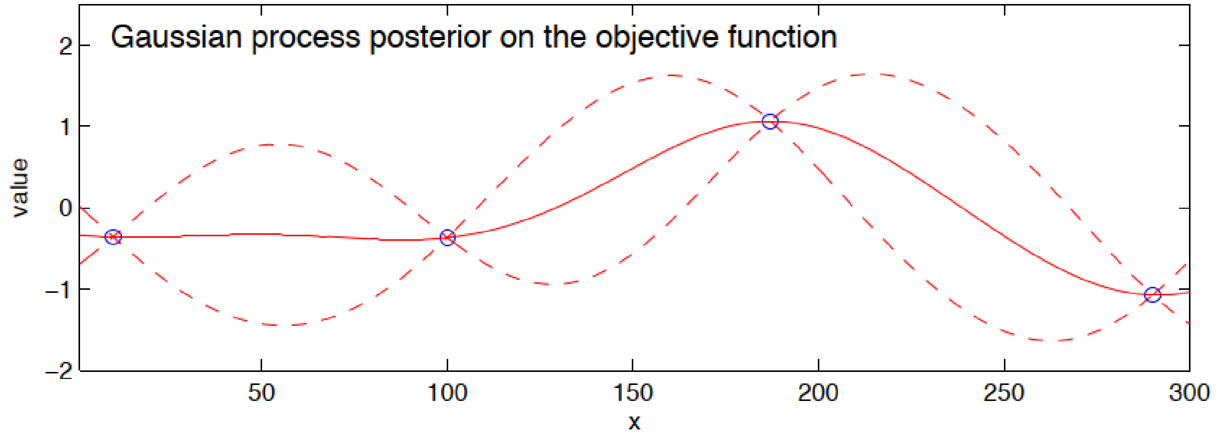
Let $f: X \rightarrow \mathbb{R}$ be a real-valued function defined on a domain X (let's assume $X \subset \mathbb{R}$ here)

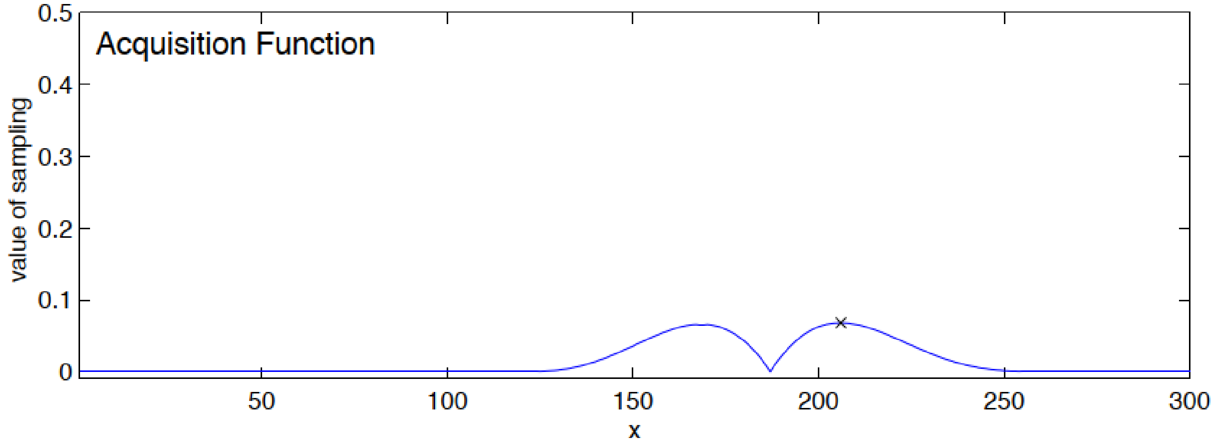




Optimizing a 1-dimensional problem

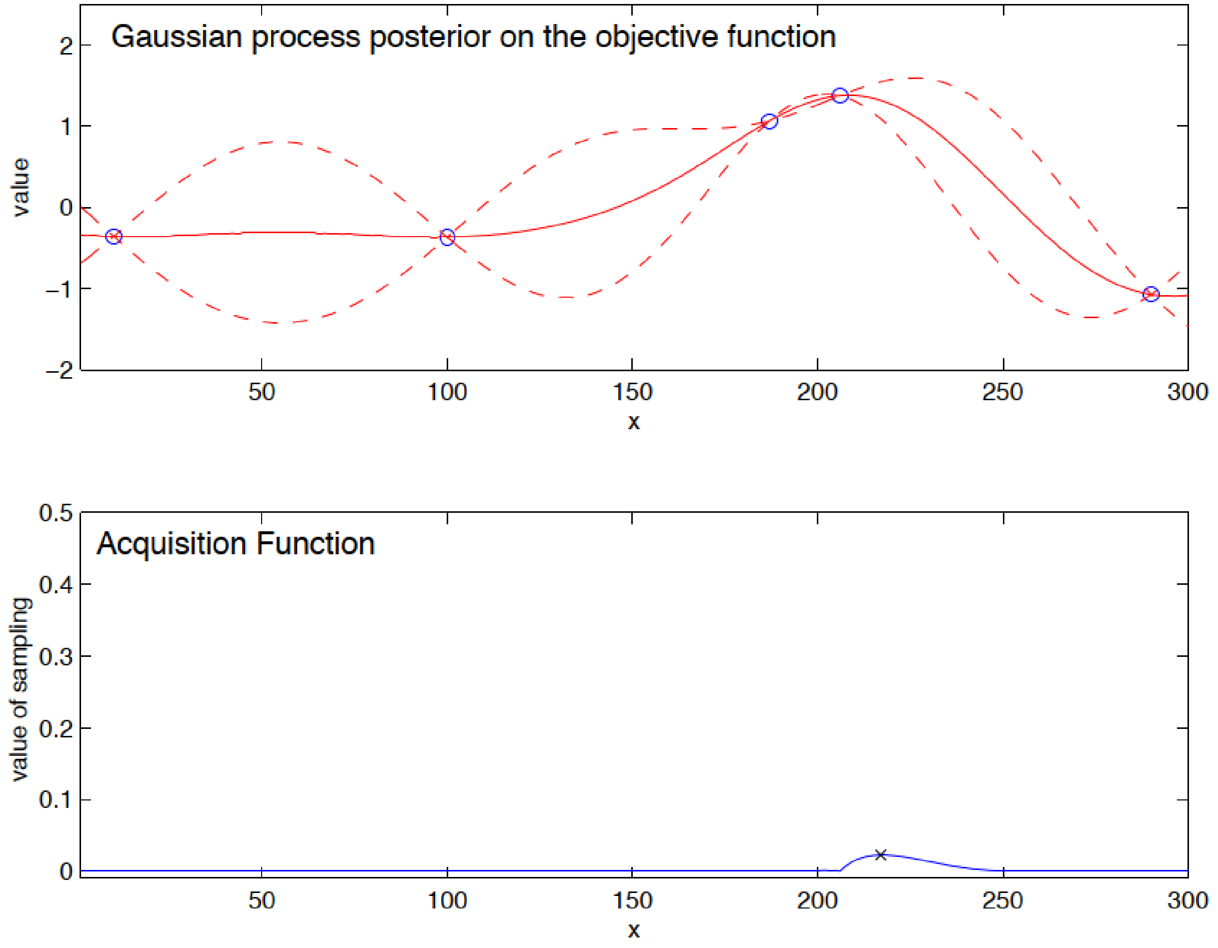
Let $f : X \rightarrow \mathbb{R}$ be a real-valued function defined on a domain X (let's assume $X \subset \mathbb{R}$ here)





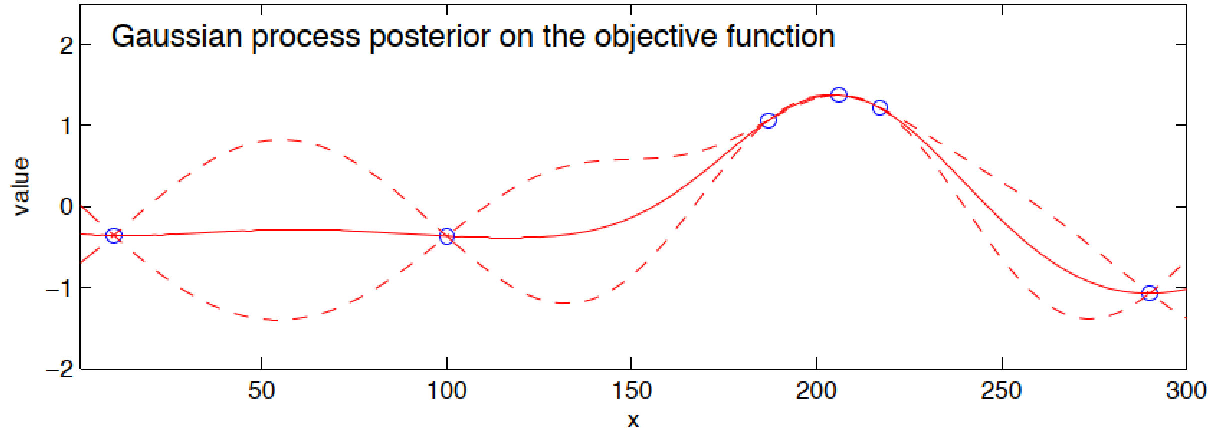
Optimizing a 1-dimensional problem

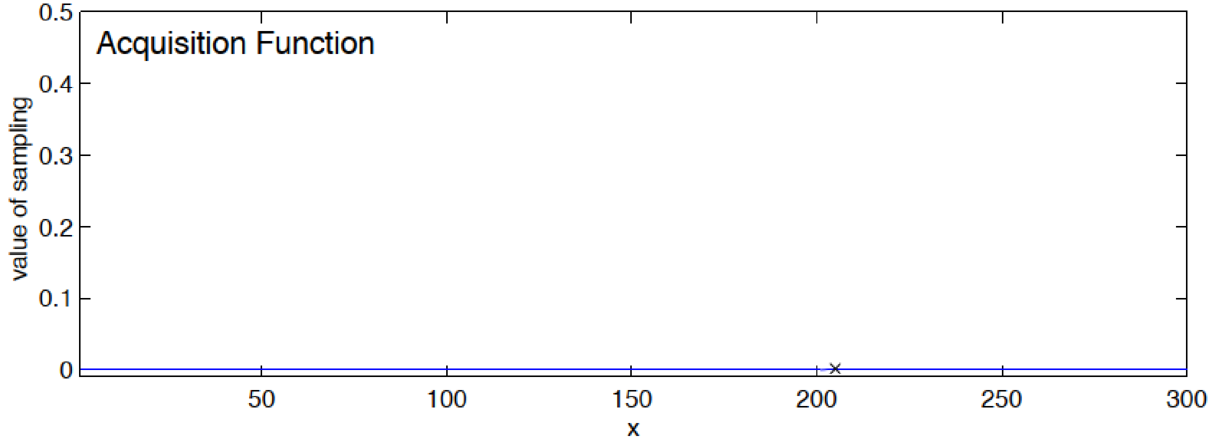
Let $f : X \to \mathbb{R}$ be a real-valued function defined on a domain X (let's assume $X \subset \mathbb{R}$ here)



Optimizing a 1-dimensional problem

Let $f: X \to \mathbb{R}$ be a real-valued function defined on a domain X (let's assume $X \subset \mathbb{R}$ here)



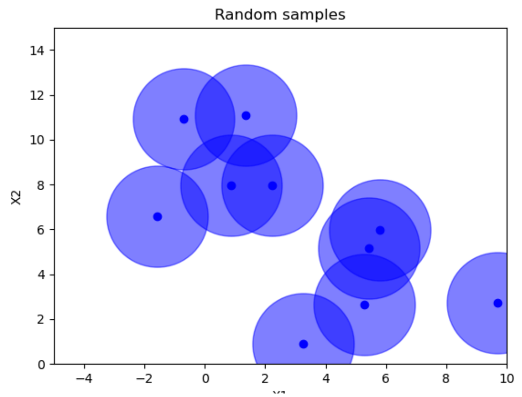


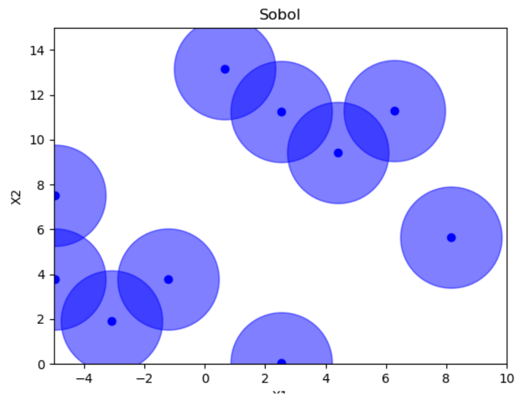
Preliminary step: Initial Sample

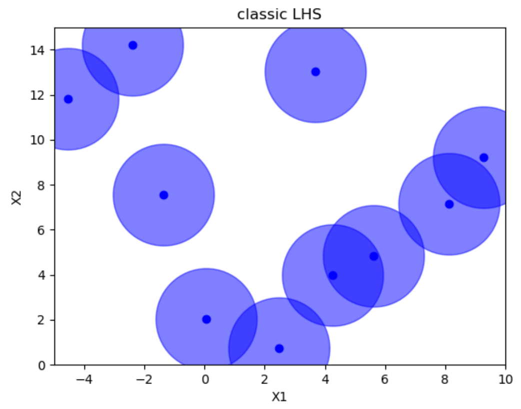
Preliminarily to the iterative procedure defining every Bayesian Optimization approach, it is common practice to sample some points in parallel, to have a first insight about the problem.

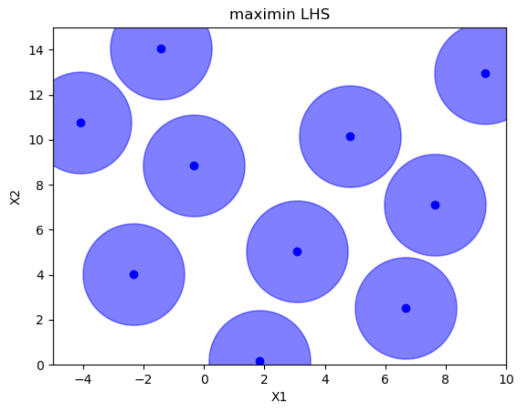
The initial sample set can be generated through many design schemes.

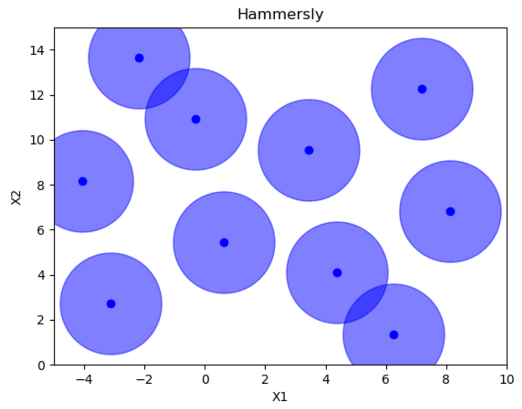
This initial step is called **Design of Experiments**.

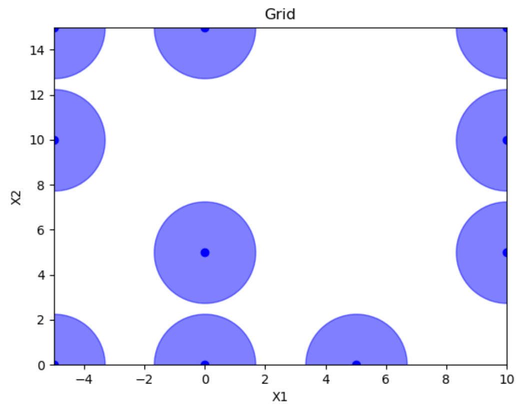












[https://scikit-optimize.github.io/stable/auto_examples/sampler/initial-sampling-method.html]

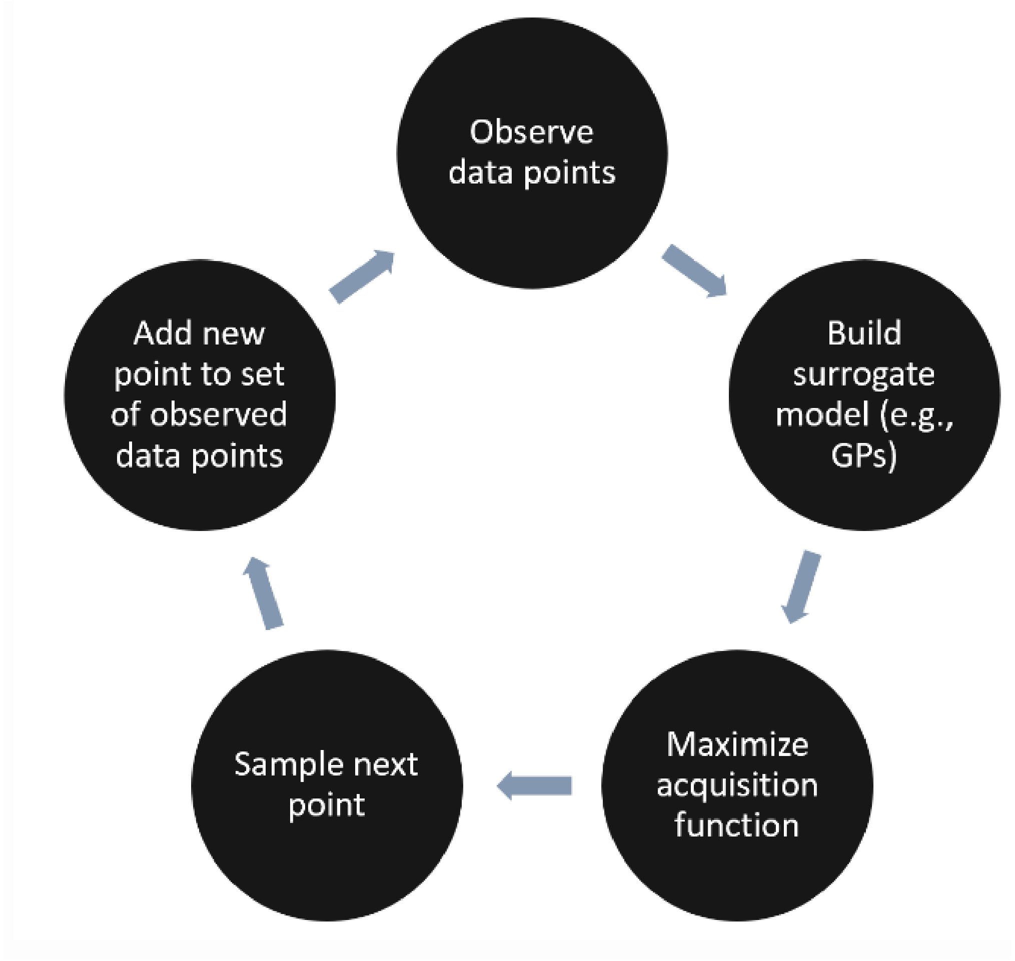
Standard logical flow of BO

Target: Find $\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x} \in X} f(\boldsymbol{x})$

Assume a Bayesian **prior** on f (usually a Gaussian process prior)

Build the **posterior** distribution

```
while (budget is not exhausted) {
    • Find x that maximizes acquisition(x,posterior)
    • Sample x & observe f(x)
    • Update the posterior distribution on f
}
```


Observe data points
Build surrogate model (e.g., GPs)
Maximize acquisition function
Sample next point
Add new point to set of observed data points

Gaussian Process Regression

Bayesian optimization usually uses **Gaussian process regression (GPR)**:

- f takes a vector x as input
- We observe f at some x and want to predict its values at other points
- GPR is a Bayesian machine learning method for doing this
- f is treated as random
- f's probability distribution is called "prior distribution" before observing data and "posterior distribution" after observing data (data = f's values at some points.

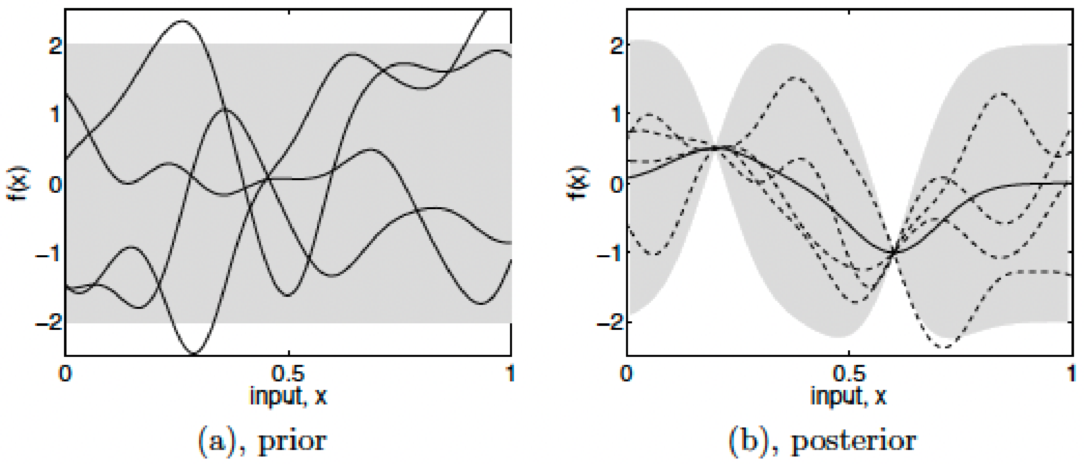


Figure 1.1: Panel (a) shows four samples drawn from the prior distribution. Panel (b) shows the situation after two datapoints have been observed. The mean prediction is shown as the solid line and four samples from the posterior are shown as dashed lines. In both plots the shaded region denotes twice the standard deviation at each input value x.

[C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006]

GPR model – Definition

Let $f: X \to \mathbb{R}$ be a real-valued function defined on a domain X,

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

A Gaussian process is completely specified by its mean function and a covariance function. We define mean function $\mu(x)$ and the covariance function $k(x, x')$ of a real process $f(x)$ as

$$\mu(x) = \mathbb{E}[f(x)]$$
$$k(x, x') = \mathbb{E}\big[\big(f(x) - \mu(x)\big)\big(f(x') - \mu(x')\big)\big],$$

and will write the Gaussian process as

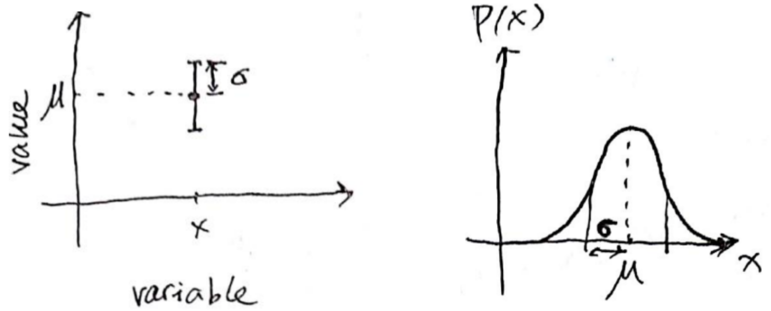
$$f(x) \sim \mathcal{GP}\big(\mu(x), k(x, x')\big).$$

In our case, the random variables represent the value of the function f(x) at location x.

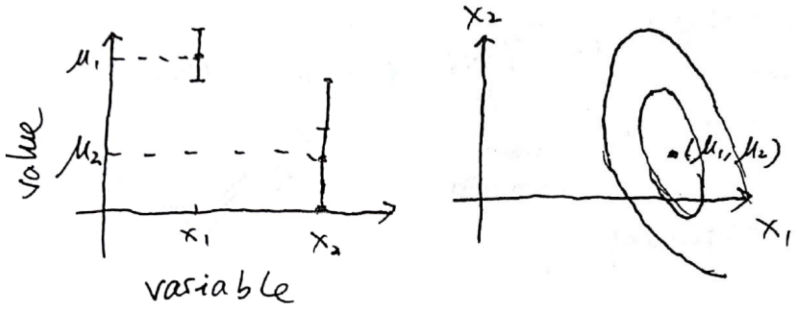
GPR model – Definition

A Gaussian process is the generalization of multivariate distribution to infinite variables. It is probability distribution over functions.

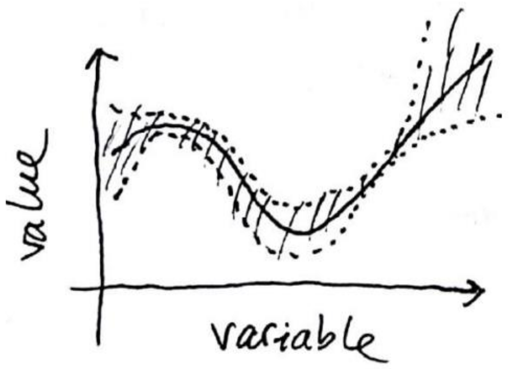
Gaussian variable
Normal distribution



Gaussian vector
Multivariate distribution



Gaussian process



GPR model – Definition

A Gaussian process is a generalization of the Gaussian probability distribution. Whereas a probability distribution describes **random variables** which are scalars or vectors (for multivariate distributions), a **stochastic process** governs the properties of functions.

- Random variable Y → get an instance y.

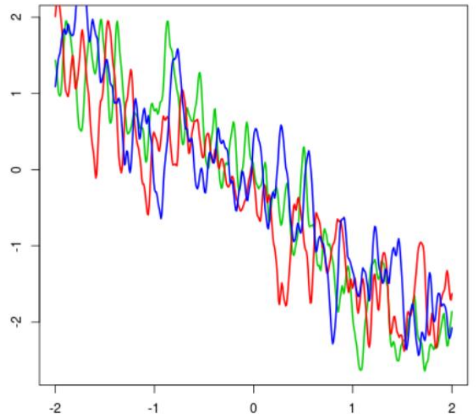
Example: random event $\omega \in \Omega$ (e.g., throw a dice) $\Longrightarrow$

- if dice ≤ 3 , $y = 1$
- if $4 \geq$ dice ≤ 5 , $y = 2$
- if dice $= 6$, $y = 3$

- Random process Y(x) → get a function y(x).

Example:
a set of RVs indexed
by x random event $\omega \in \Omega$.

We can generate several repetitions.



[A. I. J. Forrester, A. Sóbester, and A. J. Keane, Engineering Design via Surrogate Modelling - A Practical Guide. John Wiley & Sons Ltd., 2008]

How do we build a GPR model?

1. We start with a set of sample data, $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\}^T$, with observed responses $\boldsymbol{y} = \{y^{(1)}, \dots, y^{(n)}\}^T$, and we want to find an expression for a predicted value at a new point $\boldsymbol{x}$.

2. We are going to view our observed responses as if they are from a stochastic process (even though they may in fact be from a deterministic computer code). We denote this using the set of random vectors:

$$\mathbf{Y} = \begin{pmatrix} Y(\mathbf{x}^{(1)}) \\ \vdots \\ Y(\mathbf{x}^{(n)}) \end{pmatrix}.$$

3. It is a random field that has mean $\mathbf{1}\mu$, and the random variables are correlated with each other using a basis/kernel function, like for example an *exponential* basis function:

$$\operatorname{cor}[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)})] = \exp\left(-\sum_{j=1}^{k} \theta_j \mid x_j^{(i)} - x_j^{(l)} \mid^{p_j}\right)$$

How do we build a GPR model?

$$\mathrm{cor}[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)})] = \exp\left(-\sum_{j=1}^{k} \theta_j \mid x_j^{(i)} - x_j^{(l)} \mid^{p_j}\right)$$

From the correlation basis function, we can construct an $n \times n$ correlation matrix of all the observed data:

$$\boldsymbol{\Psi} = \begin{pmatrix} \mathrm{cor}[Y(\mathbf{x}^{(1)}), Y(\mathbf{x}^{(1)})] & \cdots & \mathrm{cor}[Y(\mathbf{x}^{(1)}), Y(\mathbf{x}^{(n)})] \\ \vdots & \ddots & \vdots \\ \mathrm{cor}[Y(\mathbf{x}^{(n)}), Y(\mathbf{x}^{(1)})] & \cdots & \mathrm{cor}[Y(\mathbf{x}^{(n)}), Y(\mathbf{x}^{(n)})] \end{pmatrix}$$

and a covariance matrix $\mathrm{Cov}(\mathbf{Y}, \mathbf{Y}) = \sigma^2 \boldsymbol{\Psi}.$

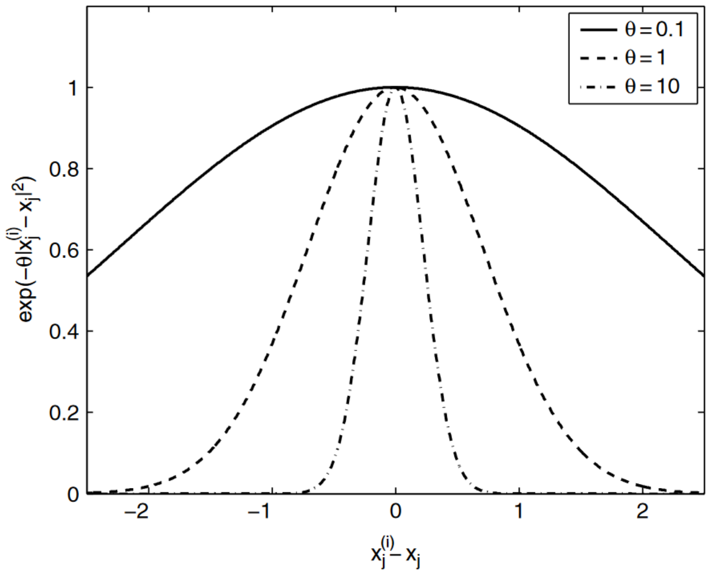
[Note that correlation is a normalized form of covariance, and some references use directly the covariance to derive the closed forms for mean and uncertainty of the model]

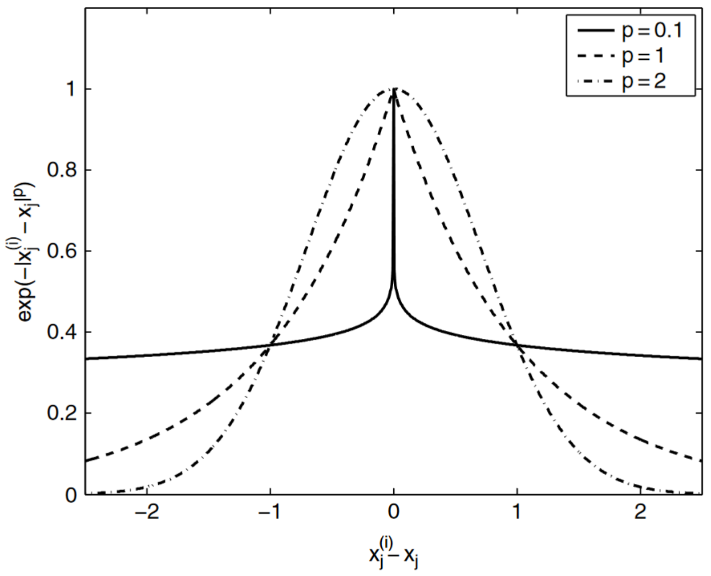
Influence of the kernel hyperparameters

$$\mathrm{cor}[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)})] = \exp\left(-\sum_{j=1}^{k} \theta_j \mid x_j^{(i)} - x_j^{(l)} \mid^{p_j}\right)$$

Correlations with varying $\boldsymbol{\theta}$ and **p:**

- **p** changes the smoothness for the correlation function
- $\boldsymbol{\theta}$ is a parameter that affects how far a sample point's influence extends





Kernel hyperparameters

$$\mathrm{cor}[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)})] = \exp\left(-\sum_{j=1}^{k} \theta_j \mid x_j^{(i)} - x_j^{(l)} \mid^{p_j}\right)$$

We now know what our correlations mean, but

1) how do we estimate the values of $\boldsymbol{\theta}$ and **p** and

2) where does our observed data **y** come in?

One answer is to choose $\boldsymbol{\theta}$ and p to maximize the likelihood of **y**:

$$L(\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(n)} \mid \mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{\sum(\mathbf{Y}^{(i)} - \mu)^2}{2\sigma^2}\right]$$

Which can be expressed in terms of the sample data as

$$L = \frac{1}{(2\pi\sigma^2)^{n/2}|\boldsymbol{\Psi}|^{1/2}} \exp\left[-\frac{(\mathbf{y} - \mathbf{1}\mu)^{\mathrm{T}}\boldsymbol{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}\right]$$

Kernel hyperparameters

$$L = \frac{1}{(2\pi\sigma^2)^{n/2}|\boldsymbol{\Psi}|^{1/2}} \exp\left[-\frac{(\mathbf{y}-\mathbf{1}\mu)^{\mathrm{T}}\boldsymbol{\Psi}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right]$$

We need to maximize the likelihood, so we can consider its natural logarithm:

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln|\boldsymbol{\Psi}| - \frac{(\mathbf{y}-\mathbf{1}\mu)^{\mathrm{T}}\boldsymbol{\Psi}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}$$

By taking the derivatives of this equation and setting them to zero, we obtain the maximum likelihood estimates (MLEs) for μ and σ^2:

$$\widehat{\mu} = \frac{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\mathbf{y}}{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\mathbf{1}},$$

$$\widehat{\sigma}^2 = \frac{(\mathbf{y}-\mathbf{1}\mu)^{\mathrm{T}}\boldsymbol{\Psi}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{n}.$$

Substitute back, remove constant terms, and optimize w.r.t. θ and p!

GPR Prediction

To predict values at unobserved locations, we need to maximize the likelihood of the sample data **AND** the prediction.

To do this, we augment the observed data with the new prediction ($\tilde{\boldsymbol{y}} = \{\boldsymbol{y}^T, \hat{y}\}^T$) and define a vector of correlations between the observed data and the new prediction:

$$\boldsymbol{\psi} = \begin{pmatrix} \mathrm{cor}[Y(\mathbf{x}^{(1)}), Y(\mathbf{x})] \\ \vdots \\ \mathrm{cor}[Y(\mathbf{x}^{(n)}), Y(\mathbf{x})] \end{pmatrix} = \begin{pmatrix} \psi^{(1)} \\ \vdots \\ \psi^{(n)} \end{pmatrix}$$

Now, we construct an augmented correlation matrix:

$$\widetilde{\boldsymbol{\Psi}} = \begin{pmatrix} \boldsymbol{\Psi} & \boldsymbol{\psi} \\ \boldsymbol{\psi}^{\mathrm{T}} & 1 \end{pmatrix}$$

GPR Prediction

The ln-likelihood of the augmented data is

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\widehat{\sigma}^2) - \frac{1}{2}\ln|\widetilde{\boldsymbol{\Psi}}| - \frac{(\widetilde{\mathbf{y}} - \mathbf{1}\widehat{\mu})^{\mathrm{T}}\widetilde{\boldsymbol{\Psi}}^{-1}(\widetilde{\mathbf{y}} - \mathbf{1}\widehat{\mu})}{2\widehat{\sigma}^2}$$

Here, only the last term depends on $\hat{y}$, so we consider only that term and optimize (substituting $\tilde{y}$ and $\widetilde{\Psi}$)

$$\ln(L) \approx \frac{-\begin{pmatrix}\mathbf{y} - \mathbf{1}\widehat{\mu} \\ \widehat{y} - \widehat{\mu}\end{pmatrix}^{\mathrm{T}} \begin{pmatrix}\boldsymbol{\Psi} & \boldsymbol{\psi} \\ \boldsymbol{\psi}^{\mathrm{T}} & 1\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{y} - \mathbf{1}\widehat{\mu} \\ \widehat{y} - \widehat{\mu}\end{pmatrix}}{2\widehat{\sigma}^2}.$$

GPR Prediction

With some matrix calculations (inversions and multiplications) we can obtain:

$$\ln(L) \approx \left(\frac{-1}{2\hat{\sigma}^2(1-\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(\hat{y}-\hat{\mu})^2 + \left(\frac{\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1-\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(\hat{y}-\hat{\mu})$$

Which we differentiate with respect to $\hat{y}$ and set to zero:

$$\left(\frac{-1}{\hat{\sigma}^2(1-\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(\hat{y}-\hat{\mu}) + \left(\frac{\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1-\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right) = 0$$

Thus, our MLE for $\hat{y}$ is: $\boxed{\hat{y}(\mathbf{x}) = \hat{\mu} + \boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu})}$

POSTERIOR MEAN

Note that: the model is constructed in such a way that the prediction goes through all the data points (it interpolates the data):

If we make a prediction at a sampled point $x^{(i)}$, ψ is the ith column of Ψ, meaning that $\psi\Psi^{-1}$ it the ith unit vector. Hence, $\hat{y}(\boldsymbol{x}) = \hat{\mu} + y^{(i)} - \hat{\mu} = y^{(i)}$, which is exactly the observed objective function value at $x^{(i)}$.

GPR Prediction

The GPR model also permits the calculation of an estimated error in the model at the queried location $\boldsymbol{x}$.

The committed error in the prediction is denoted as $\hat{s}^2$ and, after some other calculations, it is derived to be defined as:

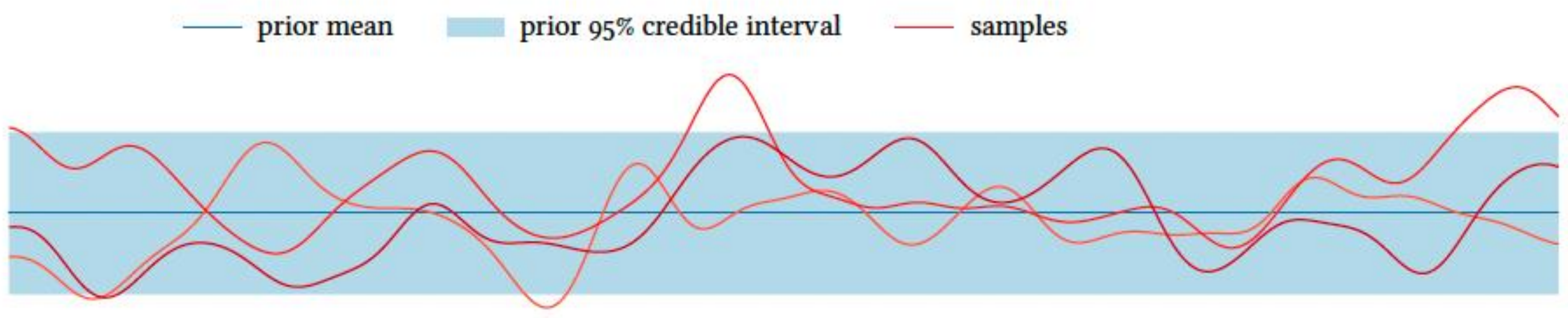
$$\boxed{\widehat{s}^2(\mathbf{x}) = \sigma^2\left[1 - \boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\boldsymbol{\psi} + \frac{1 - \mathbf{1}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}}{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\mathbf{1}}\right]}$$

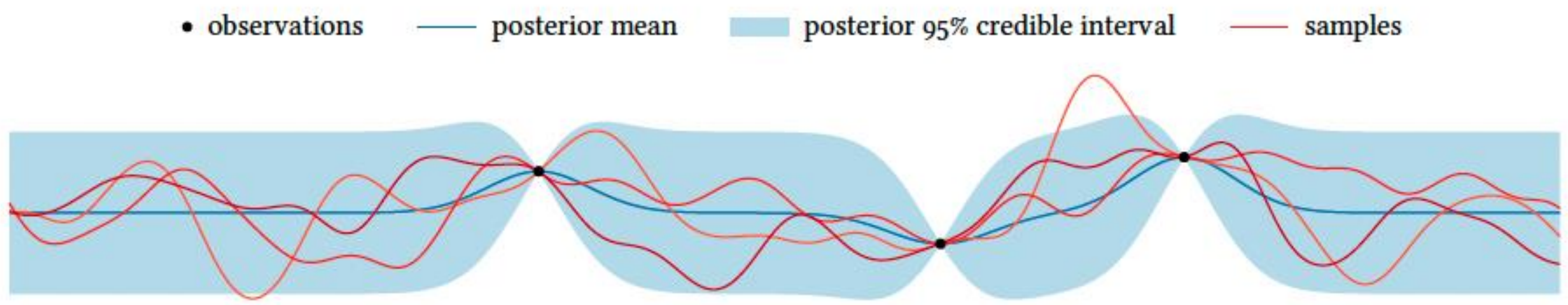
POSTERIOR VARIANCE

Based on the posterior mean and posterior variance, we have our GPR which is constructed allover the design domain and can start looking for new points to query (infill points) in an iterative fashion.

[Details can be found in: Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. (1989) Design and analysis of computer experiments. Statistical Science, 4(4), 409–423]

GPR Prediction





[R. Garnett, Bayesian Optimization Book. Accessed: Feb. 08, 2023. [Online]. Available: https://bayesoptbook.com//]

GPR Summary

- We look for an unknown response value $\hat{y}$ at the new location $\hat{\boldsymbol{x}}$
- For this, we assume that known responses and the new response have a joint Gaussian distribution with specified covariance structure

$$\begin{bmatrix} \boldsymbol{y} \\ \hat{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\boldsymbol{x}) \\ \mu(\hat{\boldsymbol{x}}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma_0} & \boldsymbol{\gamma}(\hat{\boldsymbol{x}}) \\ \boldsymbol{\gamma}(\hat{\boldsymbol{x}})^T & \Sigma_0(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}) \end{bmatrix} \right)$$

- Conditional distribution of the unknown output

$$\hat{y}|\boldsymbol{y} \sim \mathcal{N}\left(\mu(\hat{x}) + \boldsymbol{\gamma}(\hat{\boldsymbol{x}})^T \boldsymbol{\Sigma_0}^{-1}(\boldsymbol{y} - \mathbb{I}_n \boldsymbol{\mu}(\boldsymbol{x})), \Sigma_0(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}) - \boldsymbol{\gamma}(\hat{\boldsymbol{x}})^T \boldsymbol{\Sigma_0}^{-1} \boldsymbol{\gamma}(\hat{\boldsymbol{x}})\right)$$

$$\hat{\boldsymbol{y}} = \mu(\hat{x}) + \boldsymbol{\gamma}(\hat{x})^T \boldsymbol{\Sigma_0}^{-1}(\boldsymbol{y} - \mathbb{I}_n \boldsymbol{\mu}(\boldsymbol{x}))$$

Posterior mean

$$C(\boldsymbol{x}, \boldsymbol{x}') = \Sigma_0(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\gamma}(\boldsymbol{x})^T \boldsymbol{\Sigma_0}^{-1} \boldsymbol{\gamma}(\boldsymbol{x}')$$

Posterior covariance

NOTATION

Known sampled response values (training data)

Symmetric positive-definite matrix from input training data (covariance matrix)

$$\boldsymbol{\Sigma_0} = \Sigma_0\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \sigma^2 \boldsymbol{\Psi} \in \mathbb{R}^{n \times n}$$

Covariance vector $\boldsymbol{\gamma} \in \mathbb{R}^n$

$$\boldsymbol{\gamma}(\hat{\boldsymbol{x}}) = \sigma_Z^2 \left\{\psi\left(\hat{\boldsymbol{x}}, \boldsymbol{x}^{(1)}\right), \cdots, \psi\left(\hat{\boldsymbol{x}}, \boldsymbol{x}^{(n)}\right)\right\}^T$$

Alternative Kernels

Other common choices as kernels for GPs are:

Radial basis function kernel

$$k_{RBF}(x, x') = \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

Matérn kernel

$$k_M(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|x - x'|}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}|x - x'|}{\ell}\right)$$

Where

- l is a length-scale parameter
- ν is a positive parameter controlling smoothness. Three of the most popular choices are $\nu \in \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$
- K_ν is a modified Bessel function

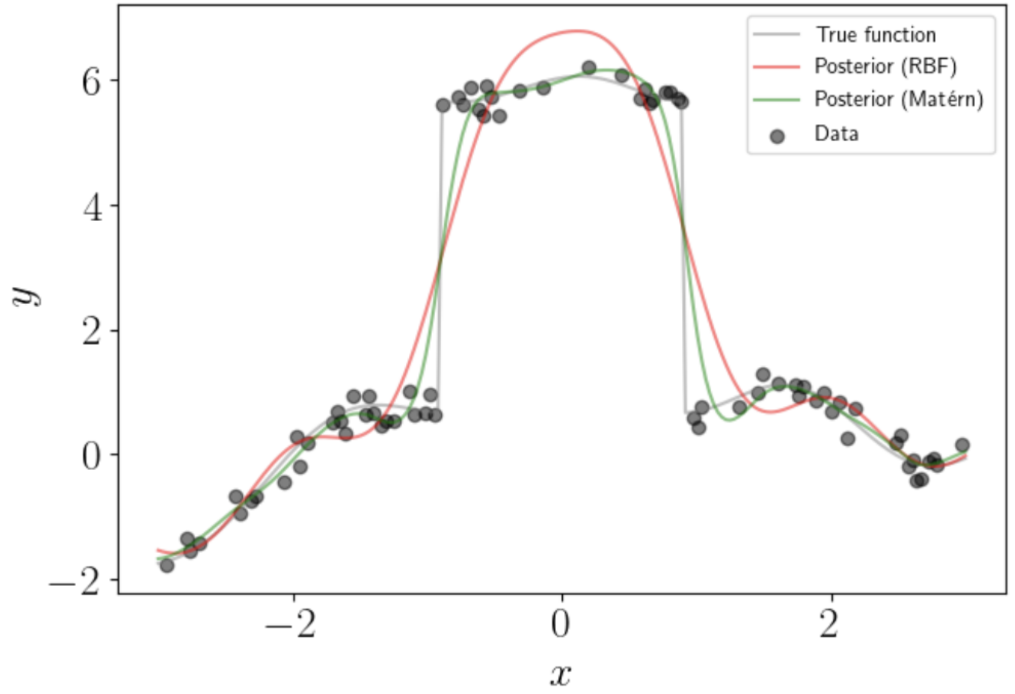
$$\left\{\begin{aligned} k_M^{1/2}(x, x') &= \exp\left(-\frac{|x - x'|}{\ell}\right) \\ k_M^{3/2}(x, x') &= \left(1 + \frac{\sqrt{3}|x - x'|}{\ell}\right)\exp\left(-\frac{\sqrt{3}|x - x'|}{\ell}\right) \\ k_M^{5/2}(x, x') &= \left(1 + \frac{\sqrt{5}|x - x'|}{\ell} + \frac{5(x - x')^2}{3\ell^2}\right)\exp\left(-\frac{\sqrt{5}|x - x'|}{\ell}\right) \end{aligned}\right.$$

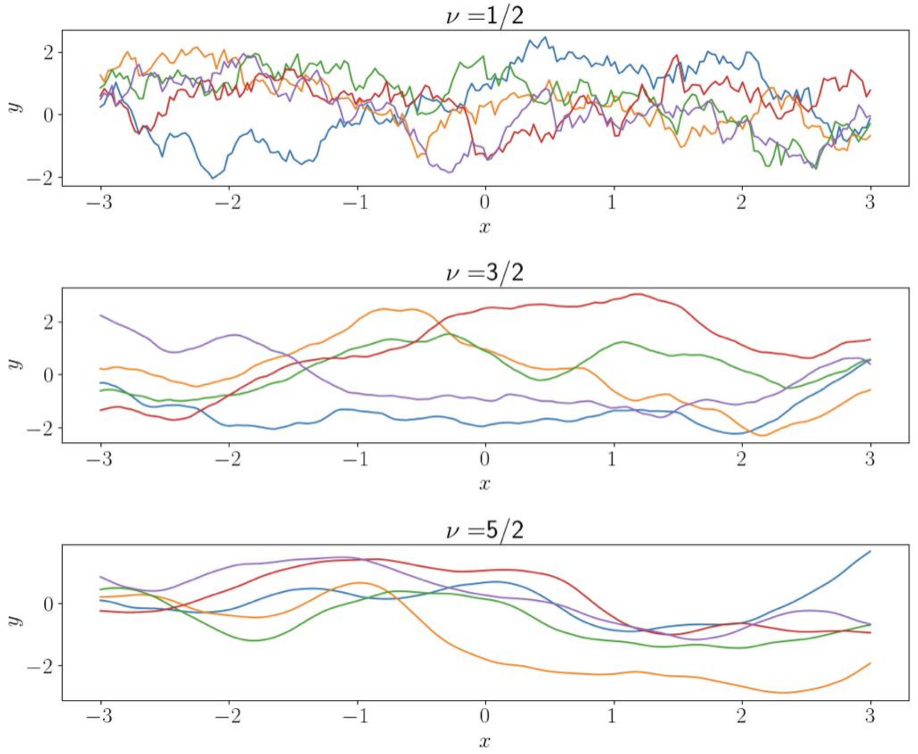
[Technical blog post by Andy Jones: https://andrewcharlesjones.github.io/journal/matern-kernels.html]

Alternative Kernels

In case of a discontinuous function, Matérn karnel is able to better capture the sharp jumps than RBF

(the plot below is for ν=3/2)

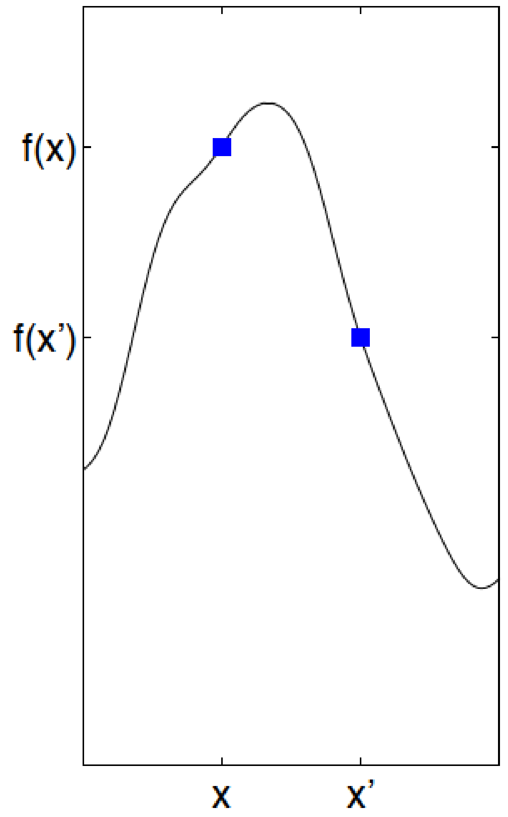




$$
\begin{cases}
k_M^{1/2}(x, x') = \exp\left(-\frac{|x-x'|}{\ell}\right) \\
k_M^{3/2}(x, x') = \left(1 + \frac{\sqrt{3}|x-x'|}{\ell}\right) \exp\left(-\frac{\sqrt{3}|x-x'|}{\ell}\right) \\
k_M^{5/2}(x, x') = \left(1 + \frac{\sqrt{5}|x-x'|}{\ell} + \frac{5(x-x')^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}|x-x'|}{\ell}\right)
\end{cases}
$$

[Technical blog post by Andy Jones: https://andrewcharlesjones.github.io/journal/matern-kernels.html]

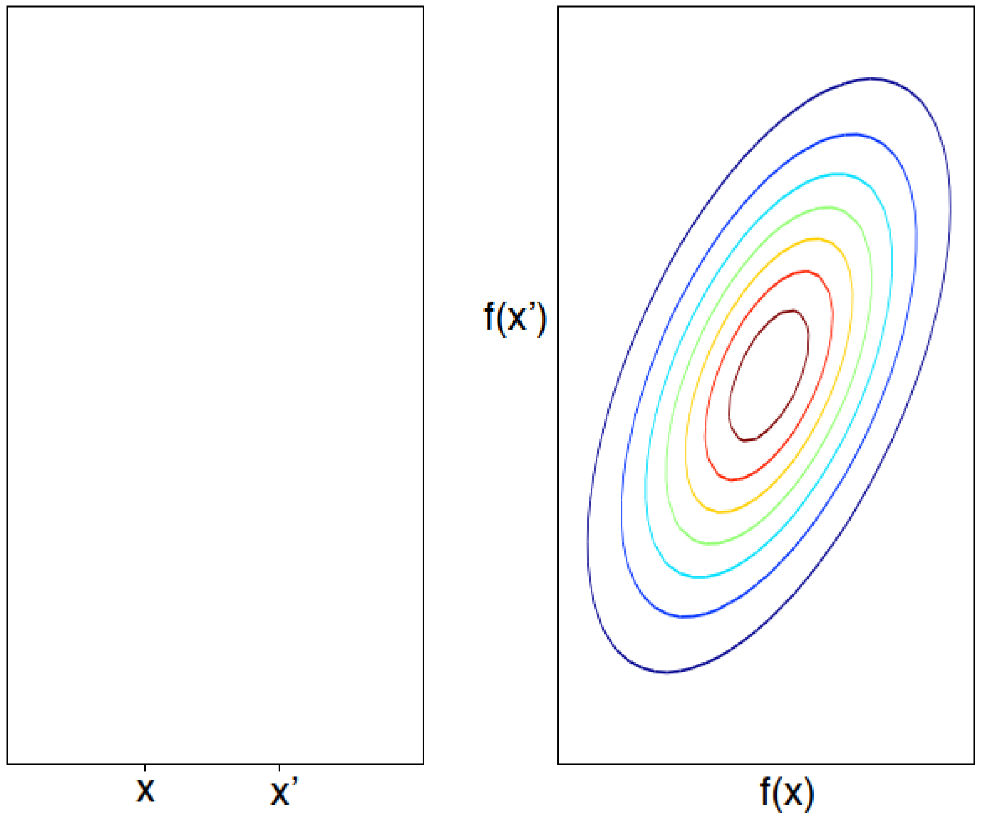
“Extra” visual example



- Fix 2 points x and x'
- Consider the vector [f(x),f(x')]
- We will understand how observing f(x') leads to a posterior distribution on f(x)

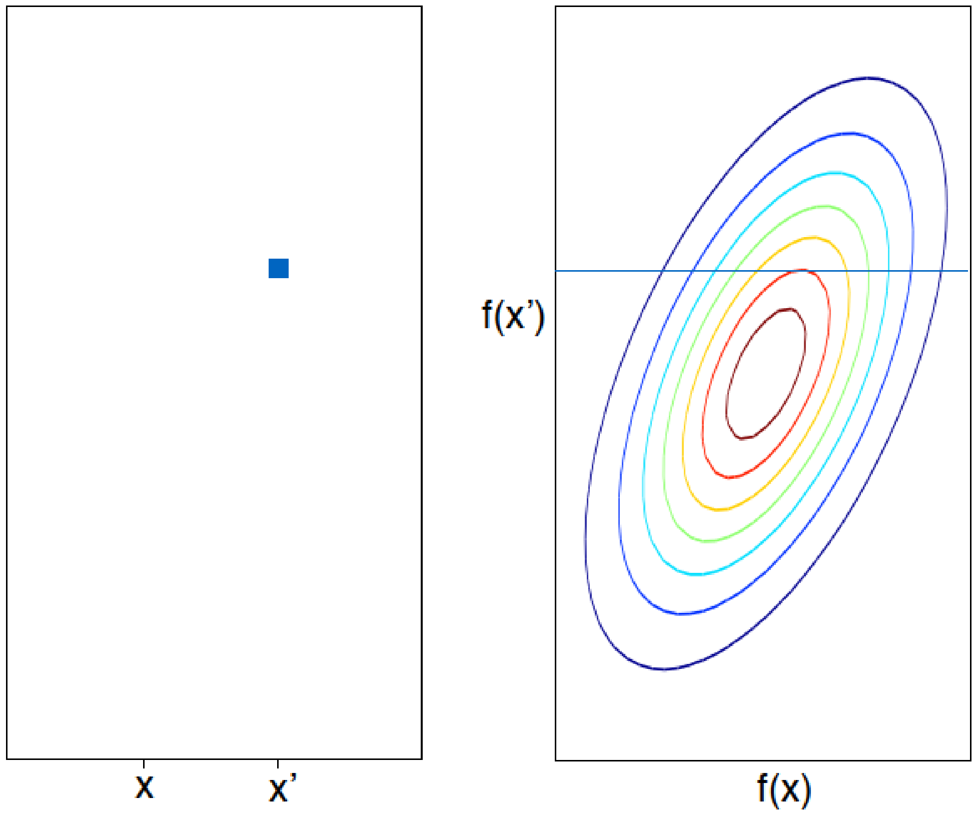
[Thanks to Peter Frazier, Eleanor and Howard Morgan Professor at Cornell University, https://people.orie.cornell.edu/pfrazier/]

“Extra” visual example



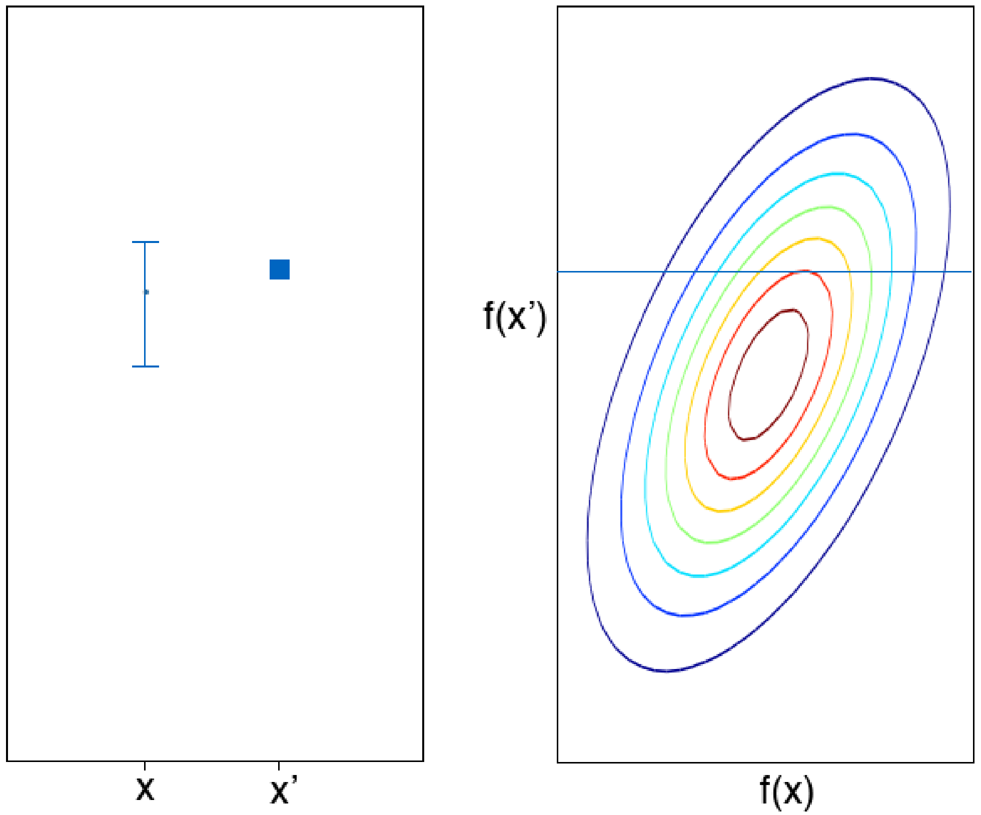
[Thanks to Peter Frazier, Eleanor and Howard Morgan Professor at Cornell University, https://people.orie.cornell.edu/pfrazier/]

“Extra” visual example



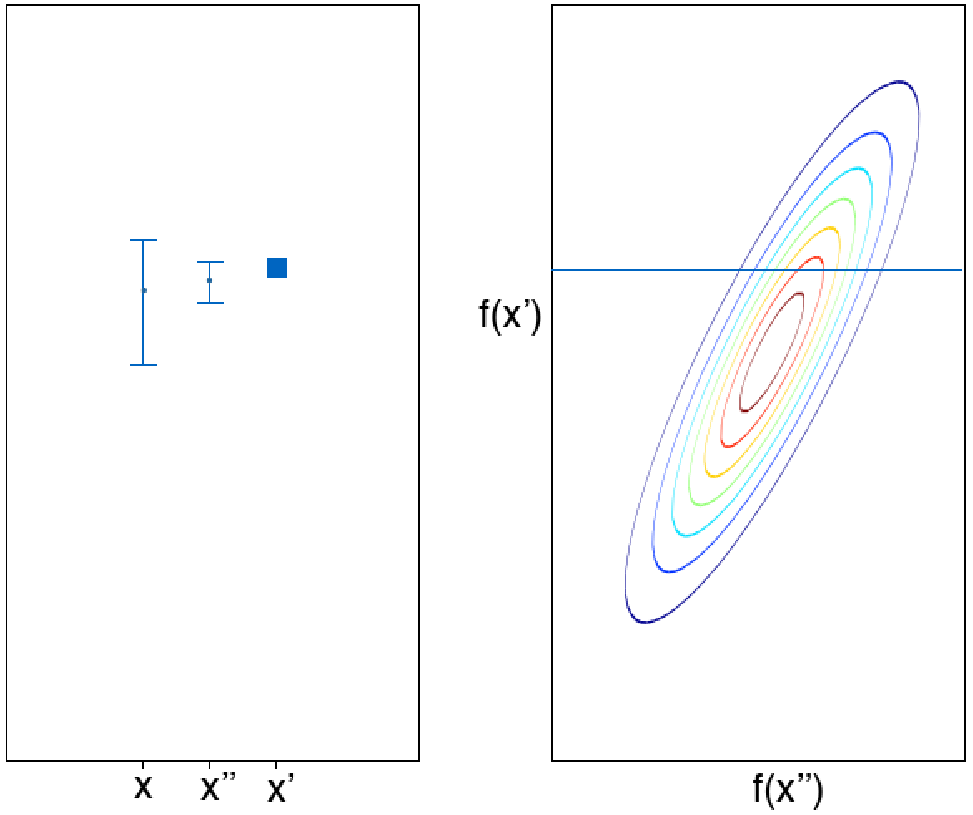
[Thanks to Peter Frazier, Eleanor and Howard Morgan Professor at Cornell University, https://people.orie.cornell.edu/pfrazier/]

“Extra” visual example



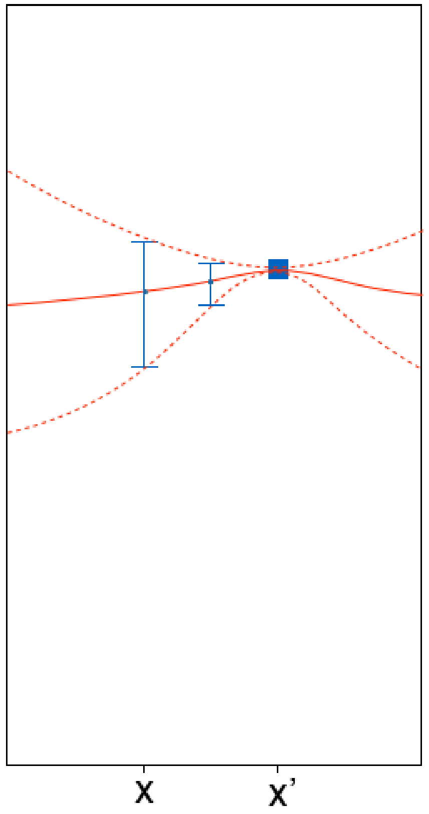
[Thanks to Peter Frazier, Eleanor and Howard Morgan Professor at Cornell University, https://people.orie.cornell.edu/pfrazier/]

“Extra” visual example



[Thanks to Peter Frazier, Eleanor and Howard Morgan Professor at Cornell University, https://people.orie.cornell.edu/pfrazier/]

“Extra” visual example



- This procedure can give us the posterior on f(x) for any x
- This posterior is normal.
  The blue dot is its mean.
  Error bars give the mean +/- 2 standard deviations.
- Repeating this procedure for every point on the x-axis gives the red solid line (posterior mean) and red dashed lines (posterior credible region)

[Thanks to Peter Frazier, Eleanor and Howard Morgan Professor at Cornell University, https://people.orie.cornell.edu/pfrazier/]

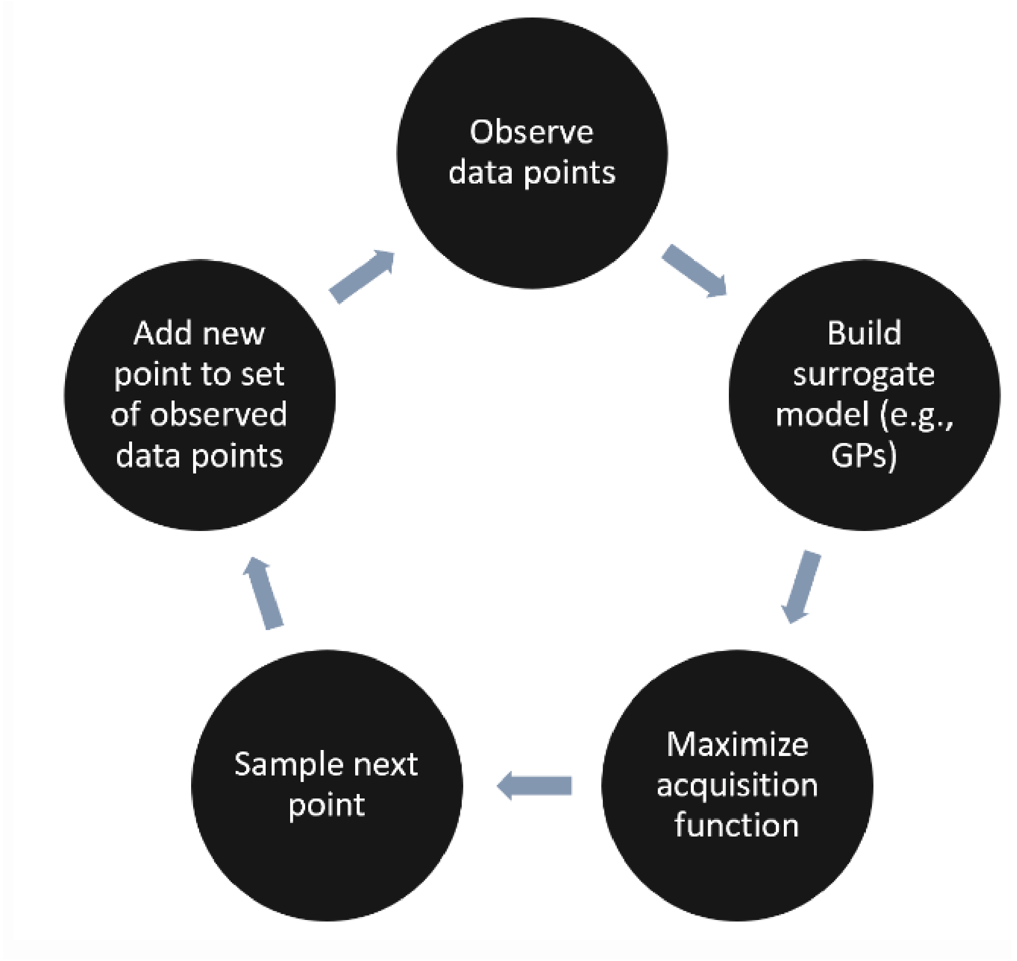
Standard logical flow of BO

Target: Find $\boldsymbol{x}^* = \operatorname{argmin}_{\boldsymbol{x} \in X} f(\boldsymbol{x})$

Assume a Bayesian **prior** on f (usually a Gaussian process prior)

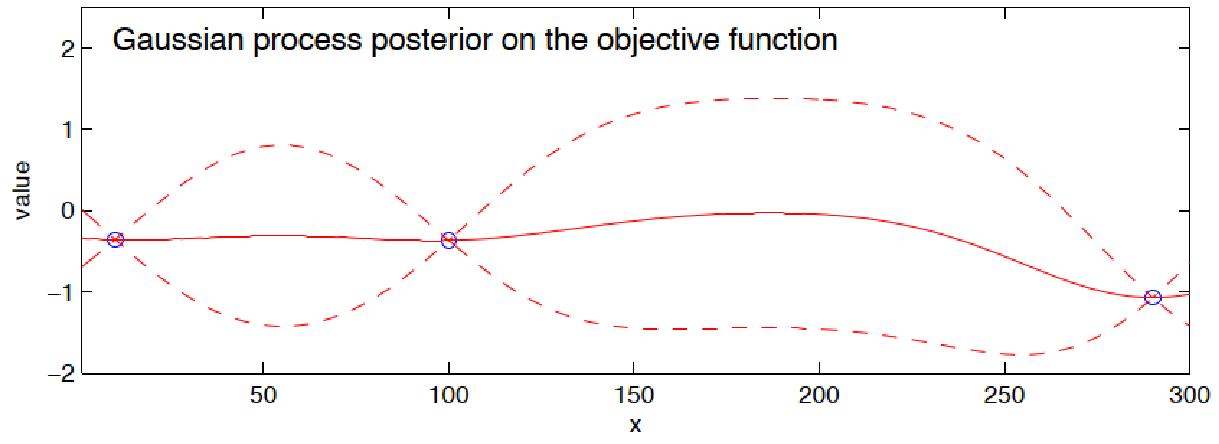
Build the **posterior** distribution

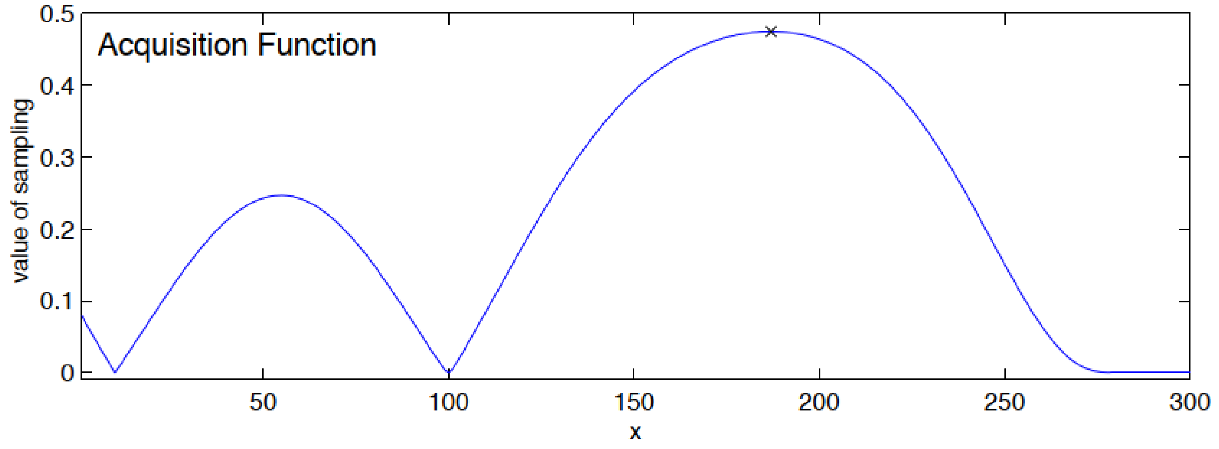
```
while (budget is not exhausted) {
    • Find x that maximizes acquisition(x,posterior)
    • Sample x & observe f(x)
    • Update the posterior distribution on f
}
```


Observe data points
Build surrogate model (e.g., GPs)
Maximize acquisition function
Sample next point
Add new point to set of observed data points

Optimizing a 1-dimensional problem

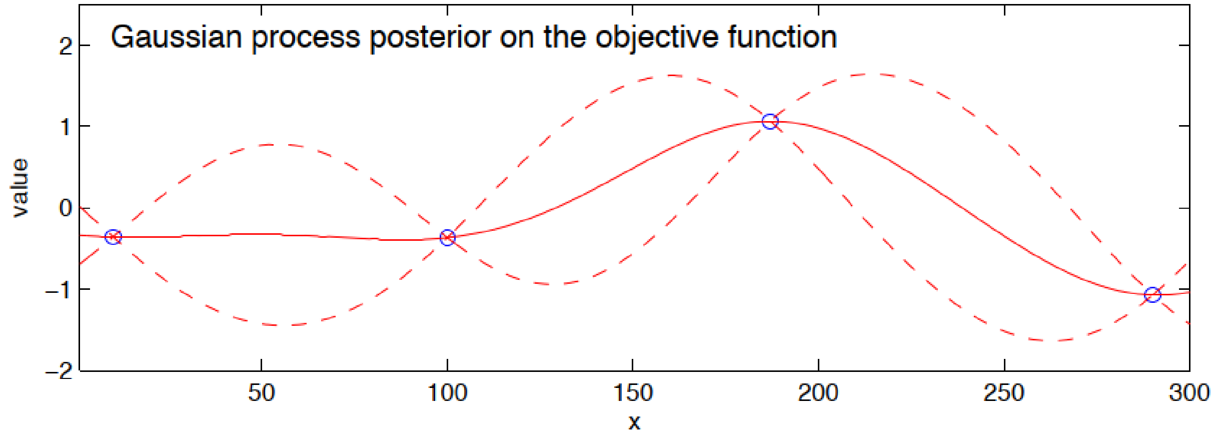
Let’s remember how the iterative and update procedure looks like:

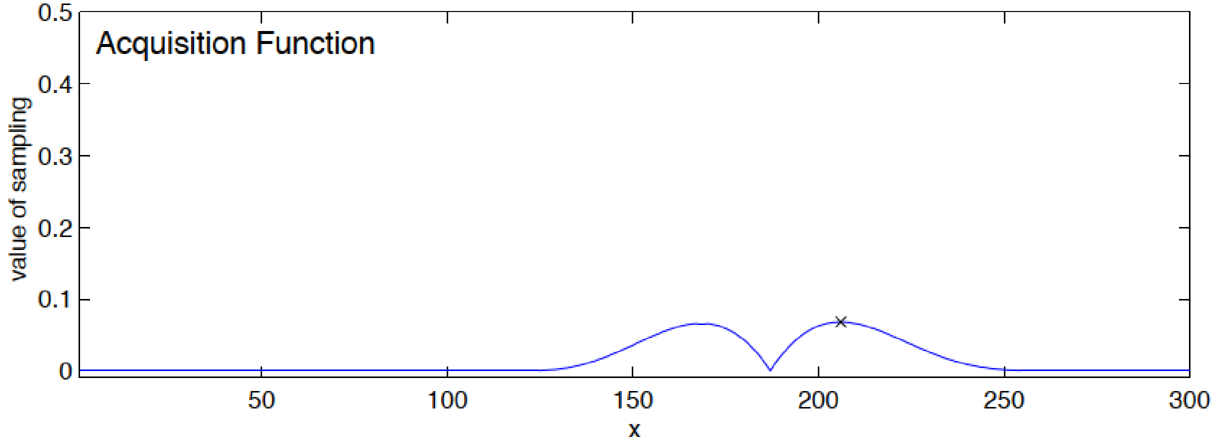




Optimizing a 1-dimensional problem

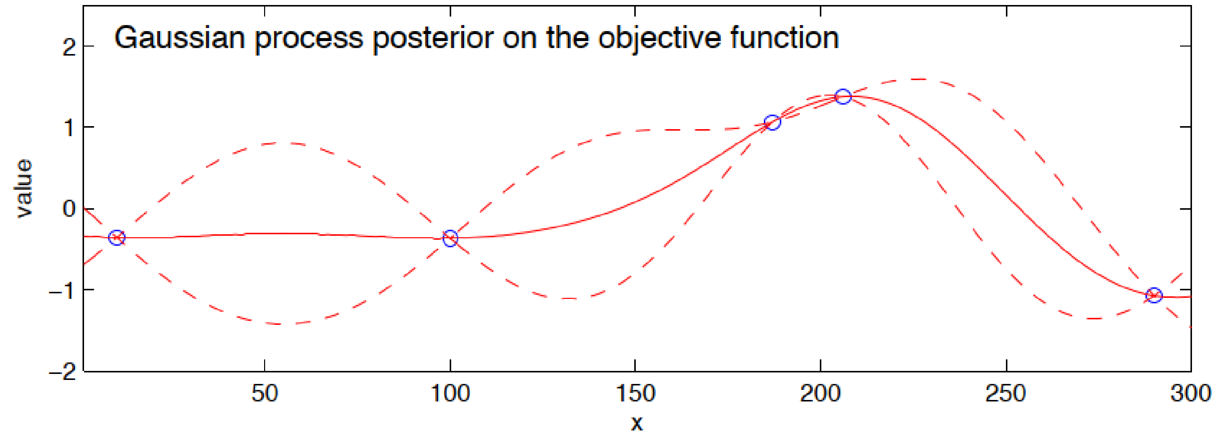
Let’s remember how the iterative and update procedure looks like:

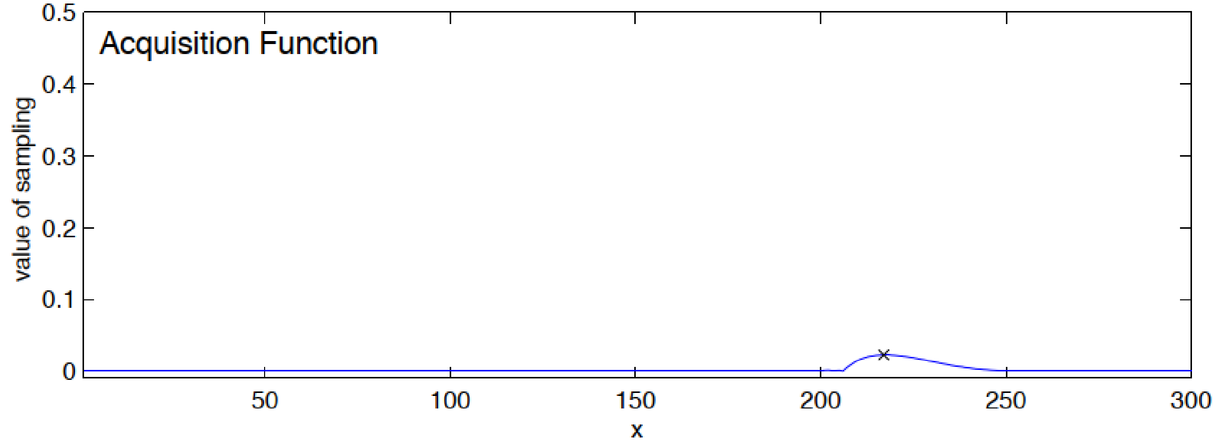




Optimizing a 1-dimensional problem

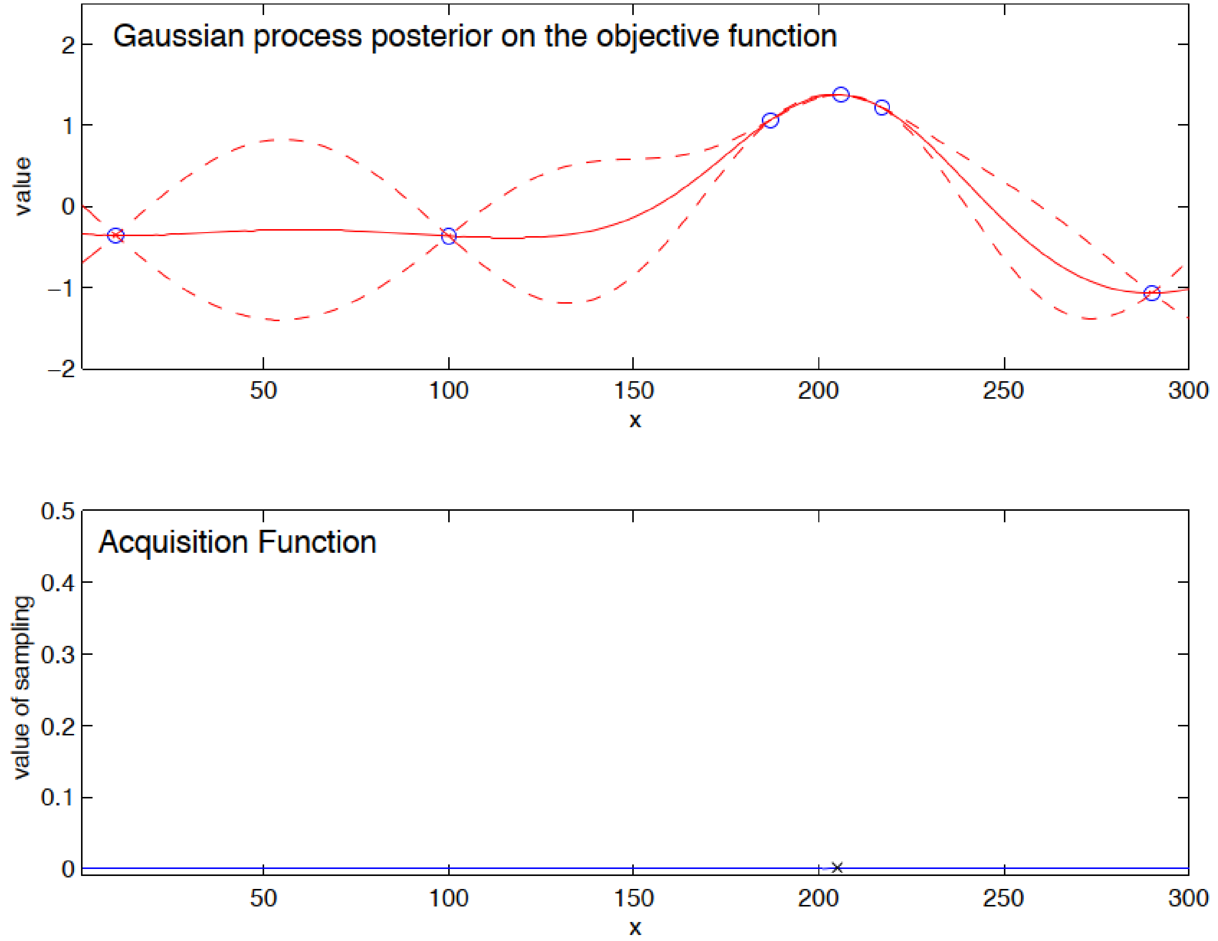
Let’s remember how the iterative and update procedure looks like:





Optimizing a 1-dimensional problem

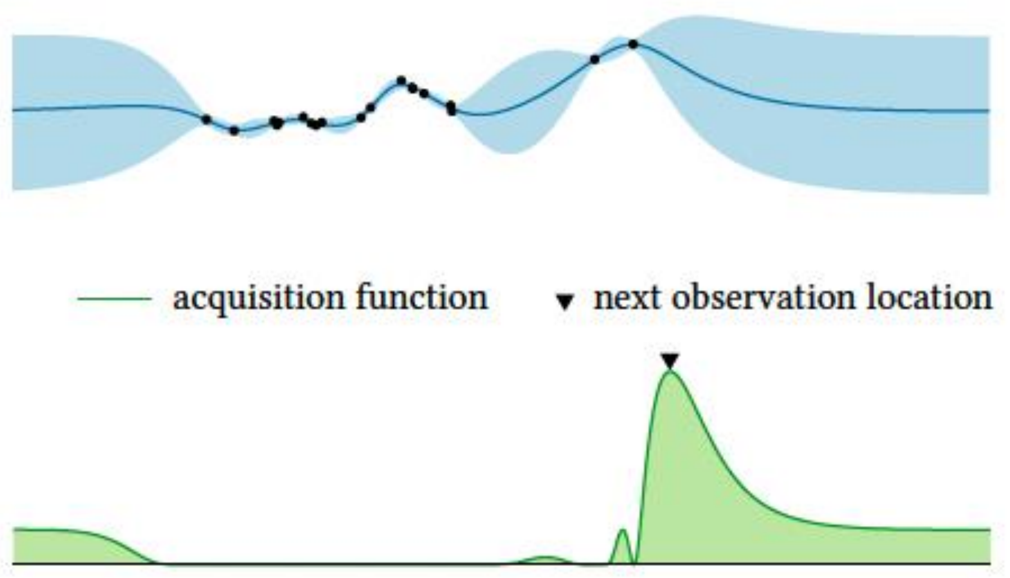
Let’s remember how the iterative and update procedure looks like:



Acquisition Function

- The iterative procedure is based on the maximization of an acquisition function (blue curve in previous slides)
- The acquisition function measures the value that would be generated by evaluation of the objective function at a new point x, based on the current posterior distribution over f
- Each time we observe f at a new point, this GPR posterior distribution is updated

- Well-known acquisition functions are:
  1. Lower/Upper confidence bound
  2. Probability of improvement
  3. Expected improvement
  4. Thompson Sampling
  5. Etc.



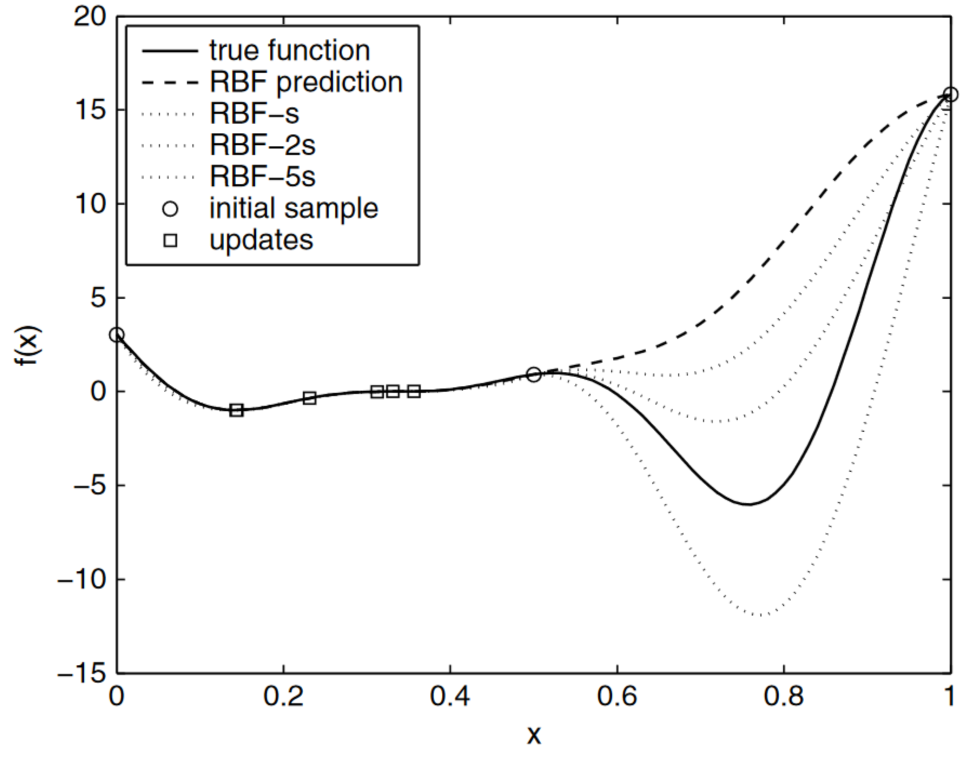
AF: Lower Confidence Bound (LCB)

The simplest way of balancing exploitation of the prediction $\hat{y}(\boldsymbol{x})$ and exploration using $\hat{s}^2(\boldsymbol{x})$ is to minimize a lower confidence bound:

$$\mathrm{LB}(\mathbf{x}) = \widehat{y}(\mathbf{x}) - A\widehat{s}(\mathbf{x}),$$

where A is a constant that controls the exploitation/exploration balance.

- As $A \to 0,\ LB(\boldsymbol{x}) \to \hat{y}(\boldsymbol{x})$ (pure exploitation)
- As $A \to \infty$, the effect of $\hat{y}(\boldsymbol{x})$ becomes negligible (pure exploration)



The lower confidence bound for varying A.

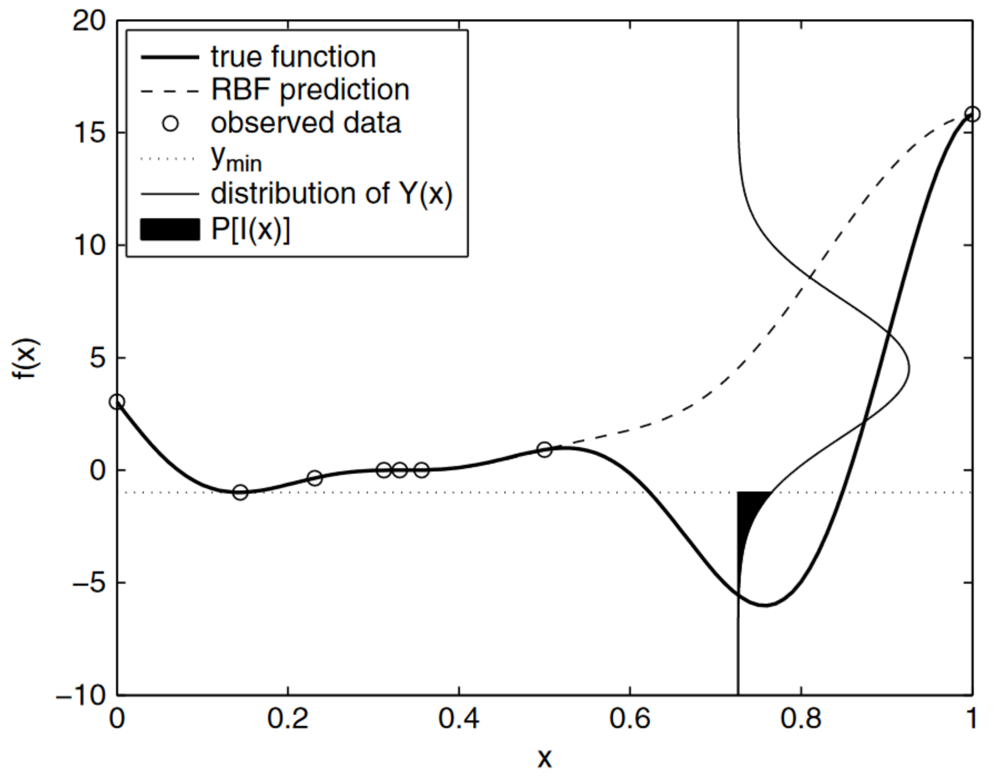
AF: Probability of Improvement (PI)

The aim is to position the infill point at a value of **x** that will lead to an improvement over the best observed so far, $y_{\min}$.

We considered the approximated value $\hat{y}(\boldsymbol{x})$ as the realization of a random variable. Hence, we can compute the probability of improvement upon $y_{\min}$,
$I = y_{\min} - Y(\boldsymbol{x})$:

$$P[I(\mathbf{x})] = \frac{1}{\hat{s}\sqrt{2\pi}} \int_{-\infty}^{0} \mathrm{e}^{-[I-\hat{y}(\mathbf{x})]^2/(2s^2)} \mathrm{d}I$$

In the figure, we see a Gaussian distribution with variance $s^2(\boldsymbol{x})$ centered around the predicted mean value at location $\boldsymbol{x}$. The distribution represents the uncertainty in the prediction and the shaded area is the probability of improving with respect to $y_{\min}$ (dotted line).



A graphical interpretation of the probability of improvement.

AF: Expected Improvement (EI)

Instead of simply finding the probability that there will be some improvement, we can calculate the amount of improvement we expect, given the mean $\hat{\mu}(\boldsymbol{x})$ and the variance $\hat{s}^2(\boldsymbol{x})$.

$$I(\mathbf{x}) = \max(y_{min} - Y(\mathbf{x}), 0)$$

$$P[I(\mathbf{x})] = P[Y(\mathbf{x}) < y_{min}]$$

$$P[I(\mathbf{x})] = \mathbf{\Phi}\left(\frac{y_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) = \frac{1}{\hat{s}(\mathbf{x})\sqrt{2\pi}} \int_{-\infty}^{0} e^{-\frac{[I(\mathbf{x})-\hat{y}(\mathbf{x})]^2}{2\hat{s}^2(\mathbf{x})}} dI, \longrightarrow \mathbf{x}_{infill} = \underset{\mathbf{x}}{\operatorname{argmax}}\ E[I(\mathbf{x})].$$

$$E[I(\mathbf{x})] = \begin{cases} \boxed{(y_{min} - \hat{y}(\mathbf{x}))\Phi\left(\frac{y_{min}-\hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right)} + \boxed{\hat{s}(\mathbf{x})\phi\left(\frac{y_{min}-\hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right)} & \text{if } \hat{s}(\mathbf{x}) > 0 \\ 0 & \text{if } \hat{s}(\mathbf{x}) = 0 \end{cases}$$

Mean-based term promoting exploitation of low function value regions

Variance-based term to explore sparse regions

where $\Phi(.)$ and $\phi(.)$ are the cumulative distribution function and probability density function, respectively.

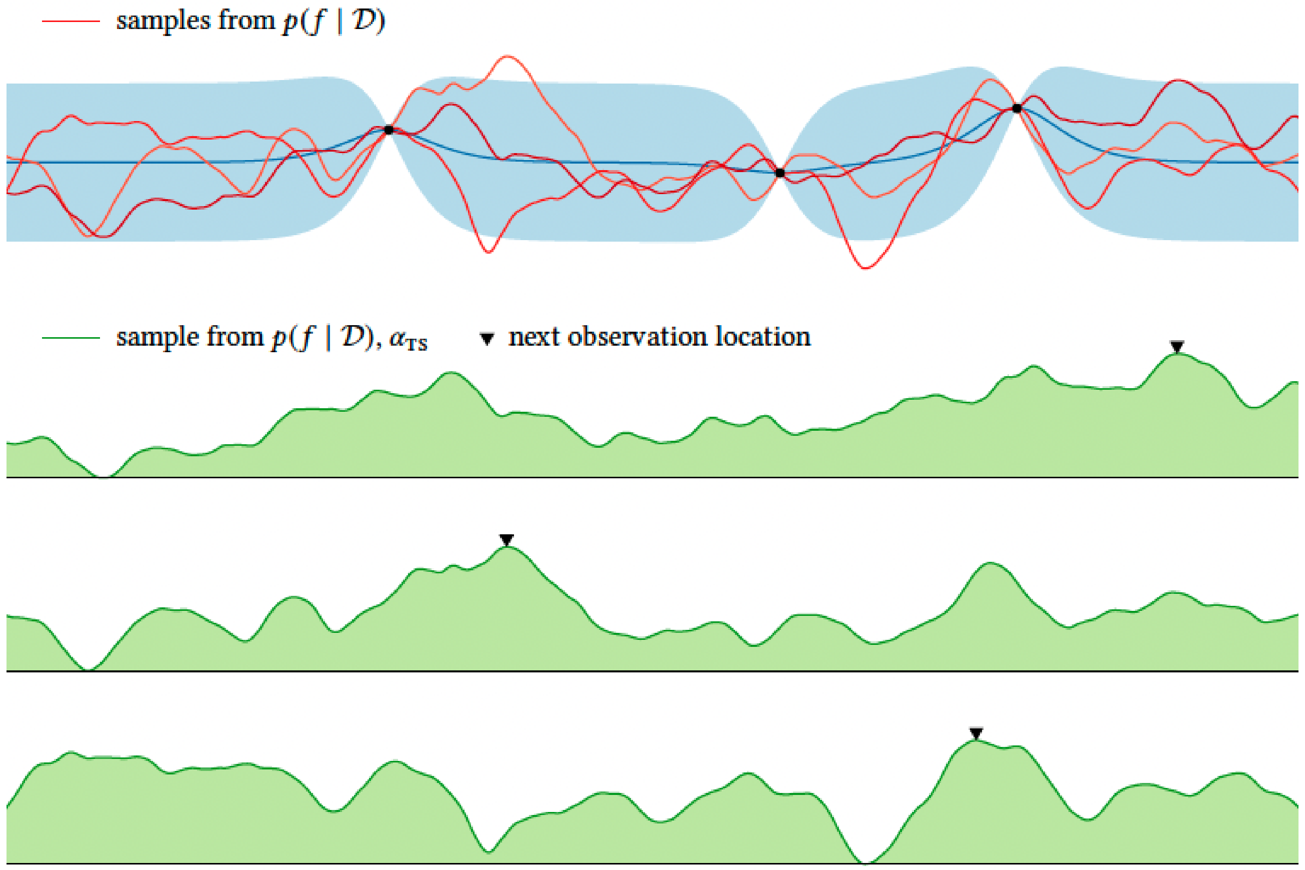
AF: Thompson Sampling (TS)

1. Thompson Sampling selects the next evaluation point by **sampling from the posterior** of the surrogate model.

2. It uses the **sampled function** to identify the point with the highest expected value and chooses it for the next evaluation.

3. Draws different samples of the GPR model and optimizes over each of them.

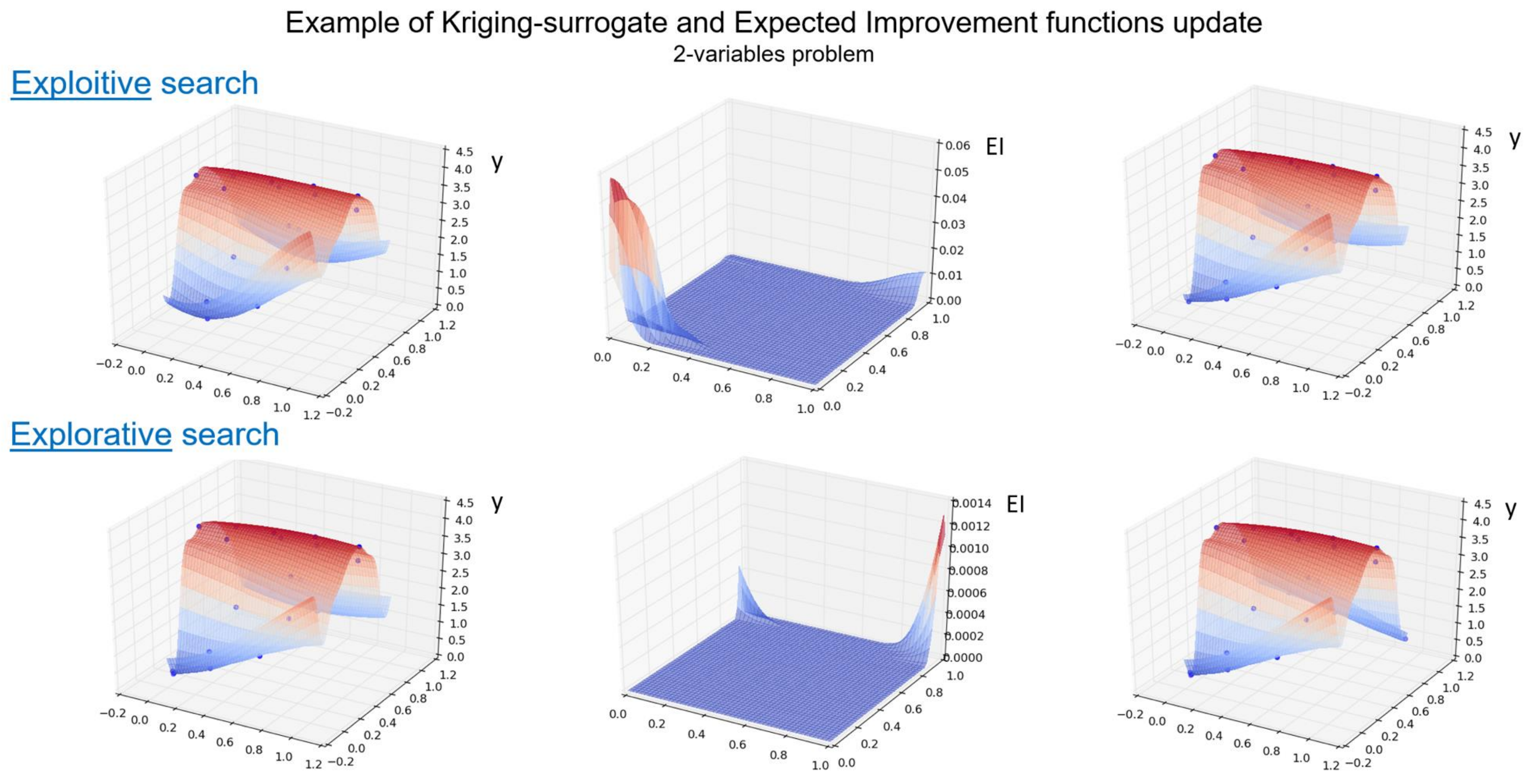
4. Variance in the suggestion is given by the repetitive nature of the procedure.



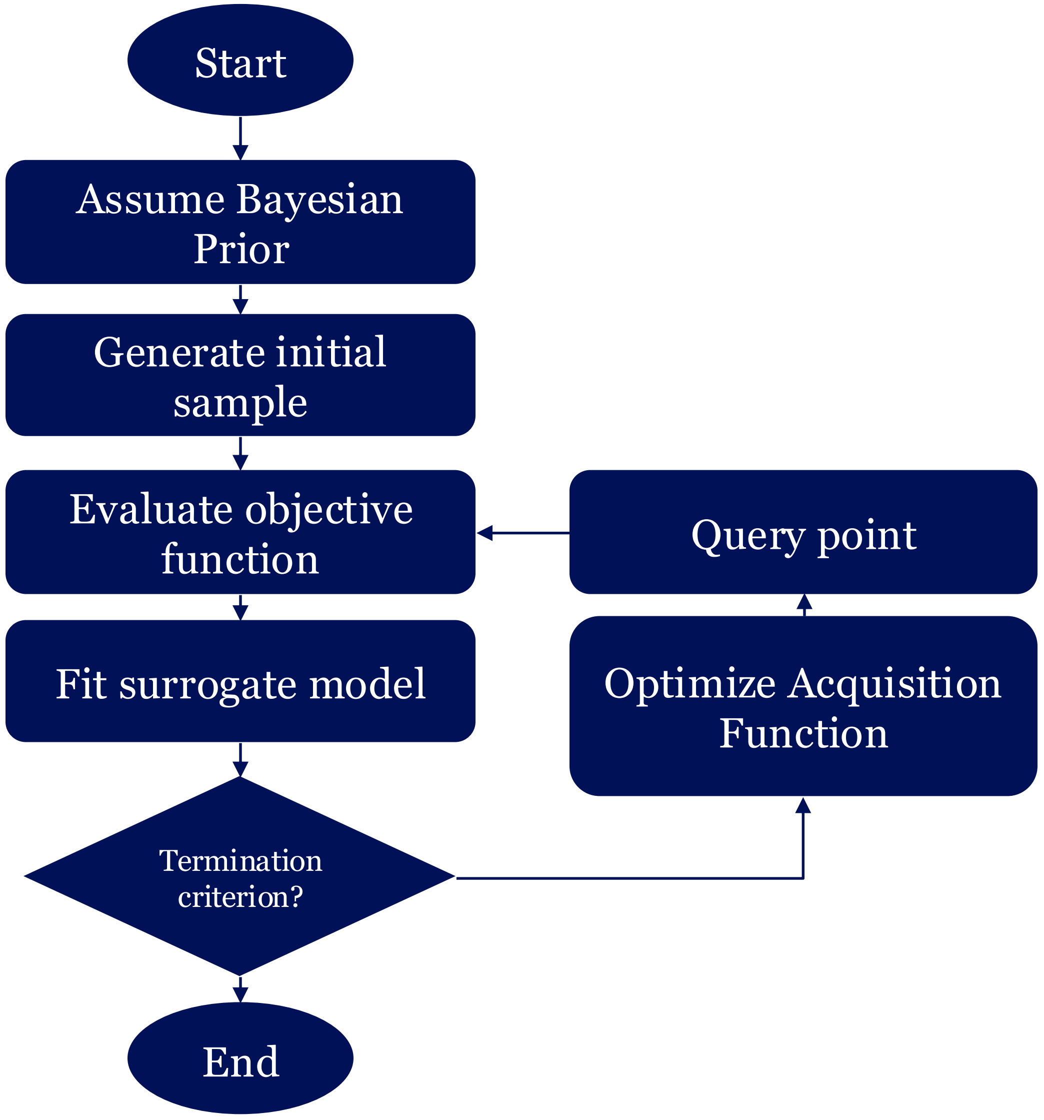
[R. Garnett, Bayesian Optimization Book. Accessed: Feb. 08, 2023. [Online]. Available: https://bayesoptbook.com//]

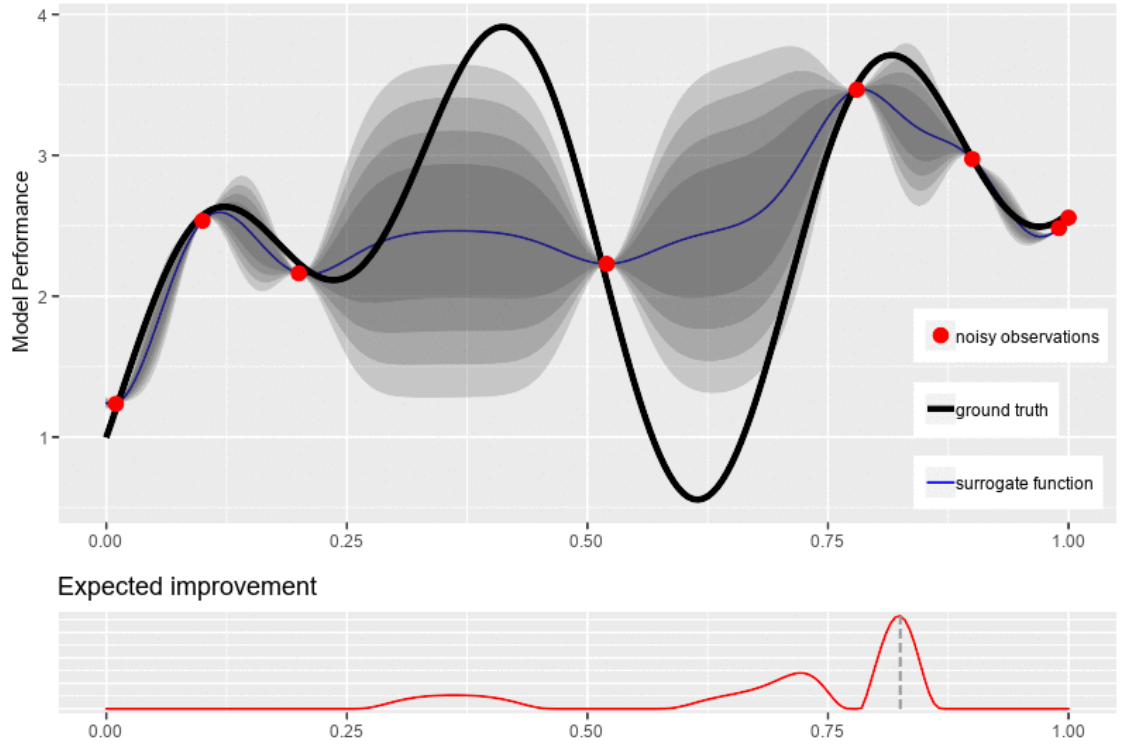
Seeked behavior in AFs…

…Good trade-off between exploitation and exploration!



Altogether – Bayesian Optimization

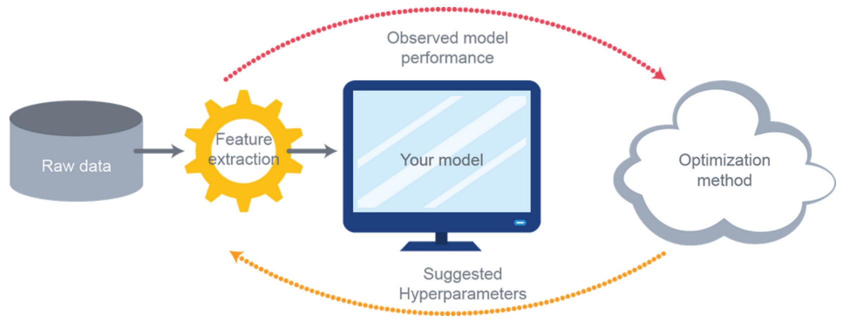


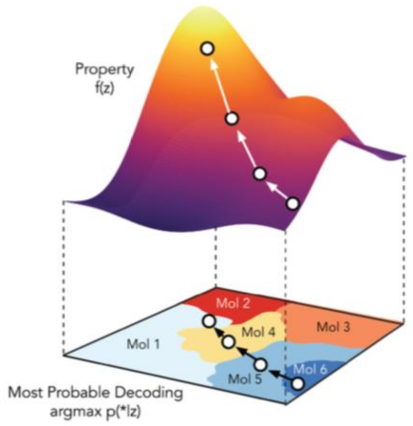


Bayesian Optimization is a big field of research!

Today, we haven't touched many subtopics:

- BO for mixed-integer/discrete spaces
- BO for noisy problems
- High-dimensional BO
- Constrained BO
- Applications of BO to real-world problems (HPO, Engineering, Chemistry…)





Thank you!

Universiteit
Leiden
The Netherlands