A Short Introduction to Bayesian Optimization

With applications to parameter tuning on accelerators

Johannes Kirschner

28th February 2018

ICFA Workshop on Machine Learning for Accelerator Control

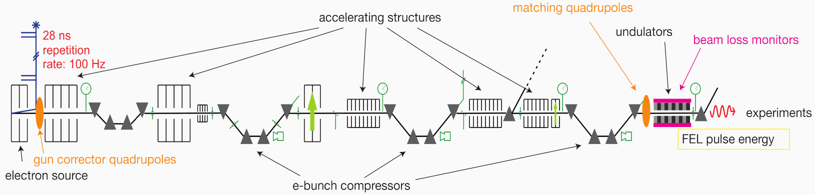
Solve

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

Example:

x = Parameter settings on accelerator

$f(x)$ = Pulse energy

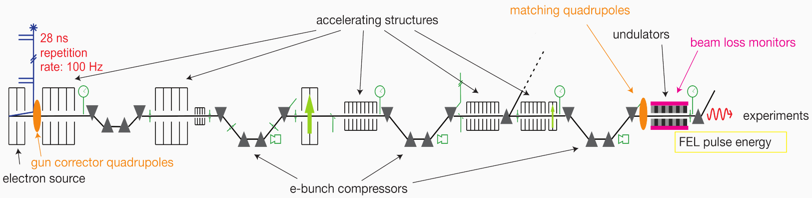


Application: Tuning of Accelerators

Example:

x = Parameter settings on accelerator

$f(x)$ = Pulse energy



Goal: Find $x^* = \arg\max_{x \in \mathcal{X}} f(x)$

... using only noisy evaluations $y_t = f(x_t) + \epsilon_t$.

Part 1)

A flexible & statistically sound model for f: Gaussian Processes

From Linear Least Squares to Gaussian Processes

Given: Measurements $(x_1, y_1), \ldots, (x_t, y_t)$.

Goal: Find statistical estimator $\hat{f}(x)$ of f.

From Linear Least Squares to Gaussian Processes

Regularized linear least squares:

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \sum_{t=1}^{T} \left(x_t^\top \theta - y_t \right)^2 + \|\theta\|^2$$

From Linear Least Squares to Gaussian Processes

Least squares regression in a Hilbert space $\mathcal{H}$:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{t=1}^{T} \big(f(x_t) - y_t\big)^2 + \|f\|_{\mathcal{H}}^2$$

From Linear Least Squares to Gaussian Processes

Least squares regression in a Hilbert space $\mathcal{H}$:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{t=1}^{T} \left(f(x_t) - y_t\right)^2 + \|f\|_{\mathcal{H}}^2$$

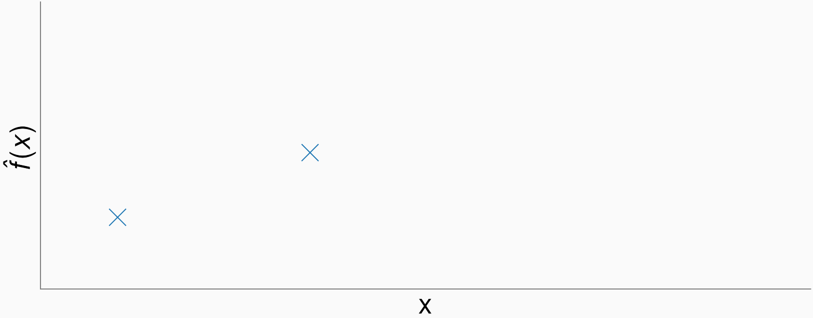
Closed form solution if $\mathcal{H}$ is a *Reproducing Kernel Hilbert Space*!

Defined by a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Example: **RBF Kernel** $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$

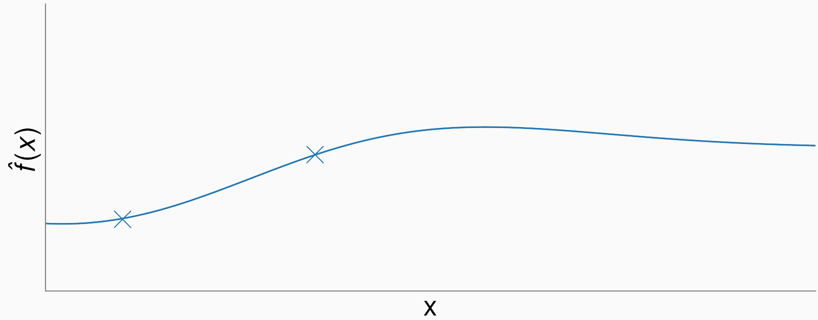
Kernel characterizes smoothness of functions in $\mathcal{H}$.

From Linear Least Squares to Gaussian Processes



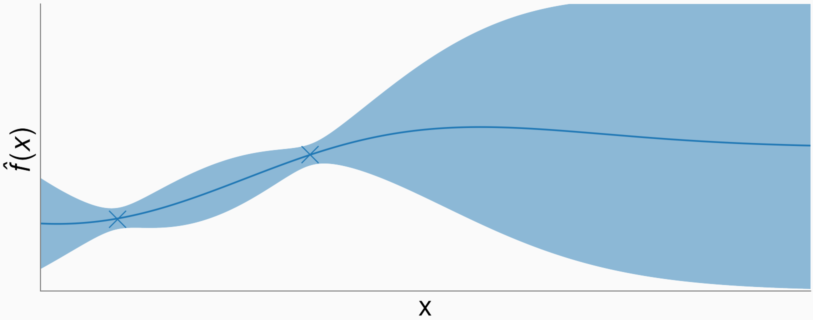
$$\hat{f} = \underset{f \in \mathcal{H}}{\arg\min} \sum_{t=1}^{T} \left(f(x_t) - y_t \right)^2 + \|f\|_{\mathcal{H}}^2$$

From Linear Least Squares to Gaussian Processes



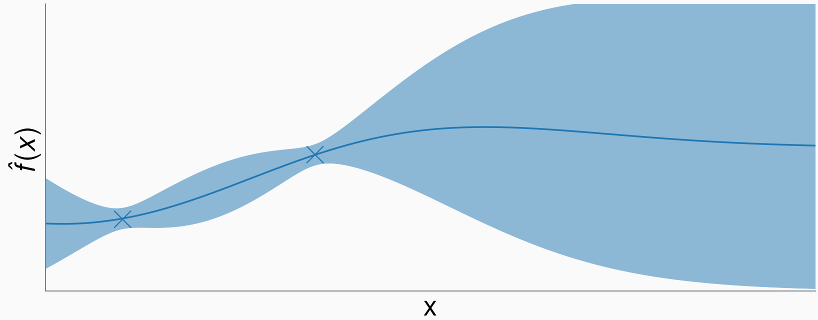
$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{t=1}^{T} \left(f(x_t) - y_t\right)^2 + \|f\|_{\mathcal{H}}^2$$

From Linear Least Squares to Gaussian Processes



$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{t=1}^{T} \left(f(x_t) - y_t\right)^2 + \|f\|_{\mathcal{H}}^2$$

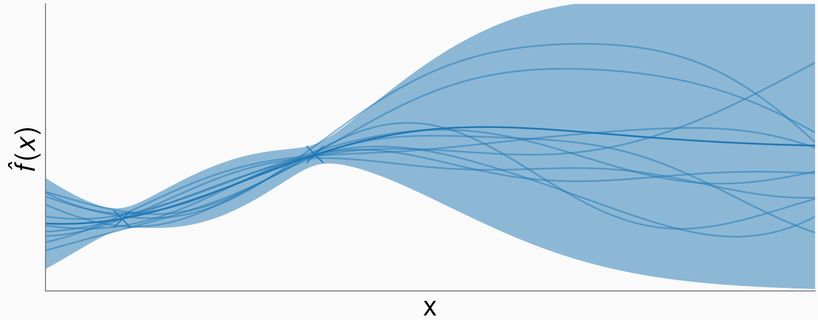
From Linear Least Squares to Gaussian Processes



Bayesian Interpretation: $\hat{f}$ is the posterior mean of a *Gaussian Process*.

A **Gaussian Process** is a **distribution over functions**, such that
- any finite collection of evaluations is multivariate normal distributed,
- the covariance structure is defined through the kernel.

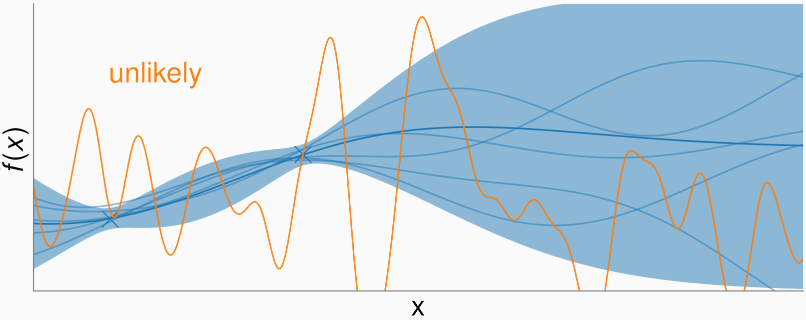
From Linear Least Squares to Gaussian Processes



Bayesian Interpretation: $\hat{f}$ is the posterior mean of a *Gaussian Process*.

A **Gaussian Process** is a **distribution over functions**, such that
- any finite collection of evaluations is multivariate normal distributed,
- the covariance structure is defined through the kernel.

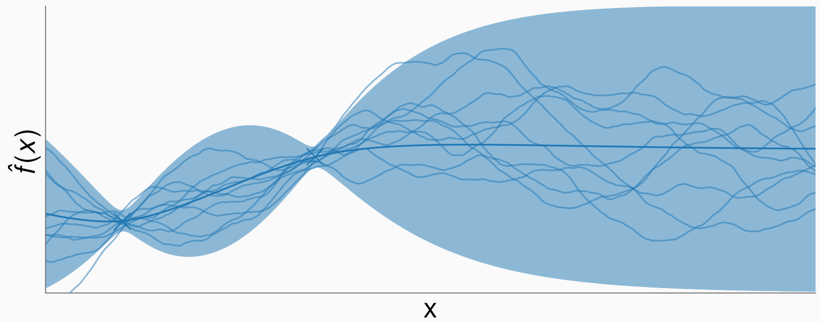
From Linear Least Squares to Gaussian Processes



Bayesian Interpretation: $\hat{f}$ is the posterior mean of a *Gaussian Process*.

A **Gaussian Process** is a **distribution over functions**, such that
- any finite collection of evaluations is multivariate normal distributed,
- the covariance structure is defined through the kernel.

From Linear Least Squares to Gaussian Processes



Bayesian Interpretation: $\hat{f}$ is the posterior mean of a *Gaussian Process*.

A **Gaussian Process** is a **distribution over functions**, such that
- any finite collection of evaluations is multivariate normal distributed,
- the covariance structure is defined through the kernel.

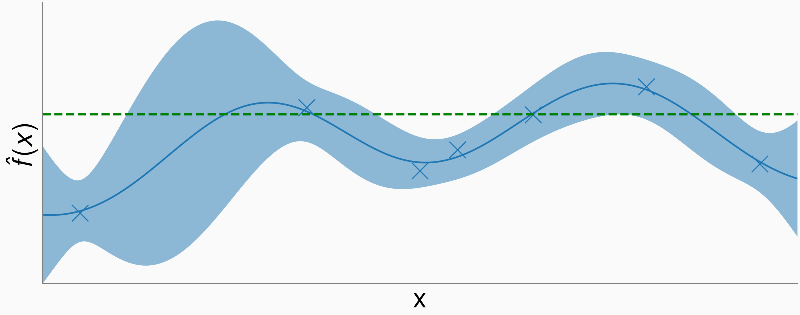
Part 2)

Bayesian Optimization Algorithms

Bayesian Optimization: Introduction

Idea: Use confidence intervals to efficiently optimize f.

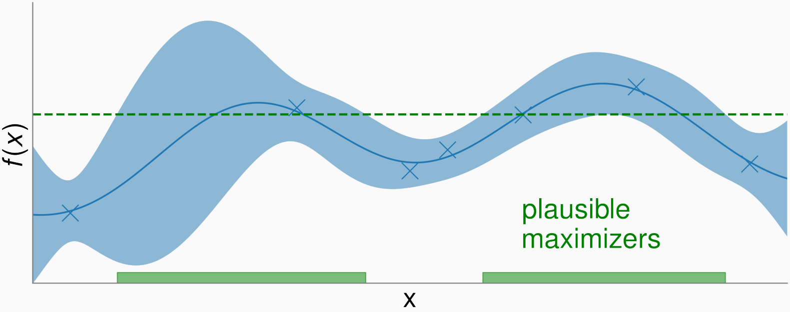
Example: Plausible Maximizers



Bayesian Optimization: Introduction

Idea: Use confidence intervals to efficiently optimize f.

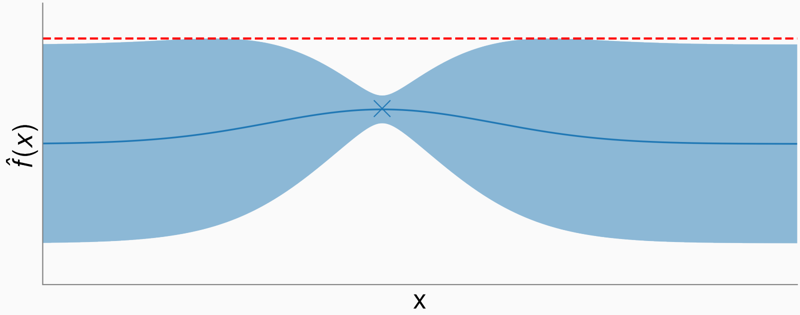
Example: Plausible Maximizers



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

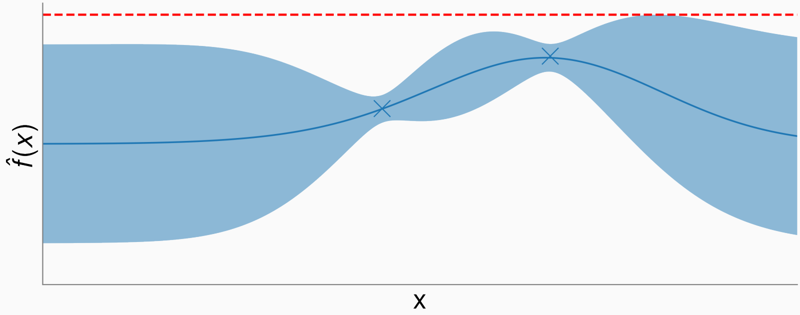
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

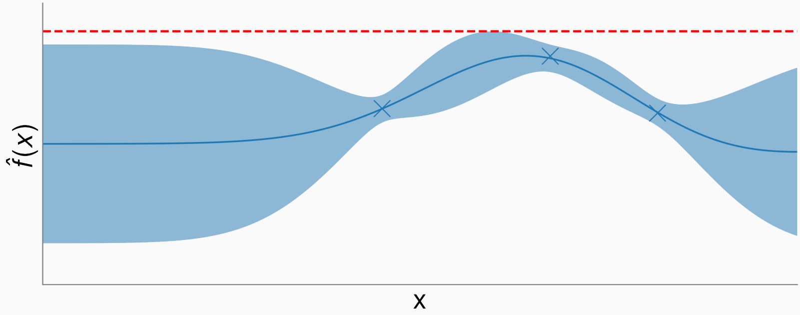
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

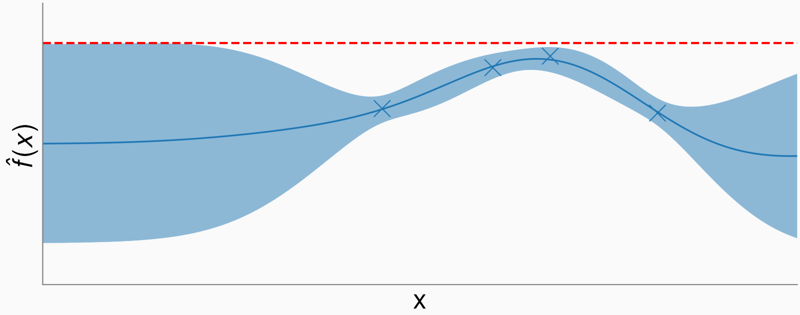
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

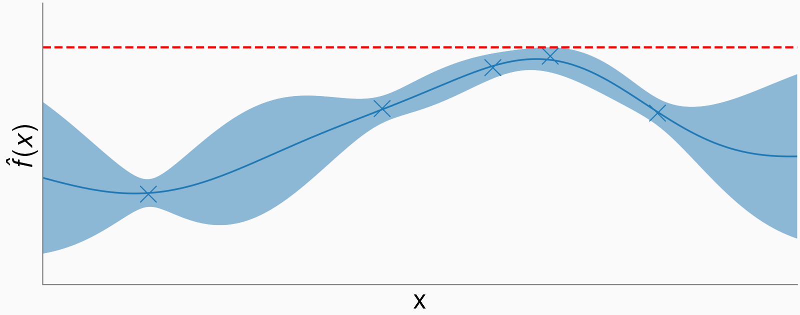
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

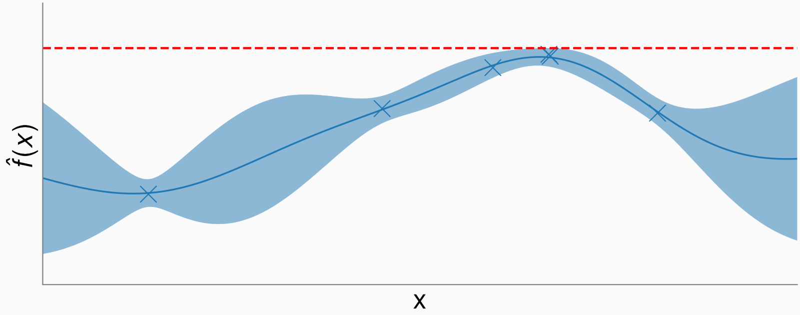
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

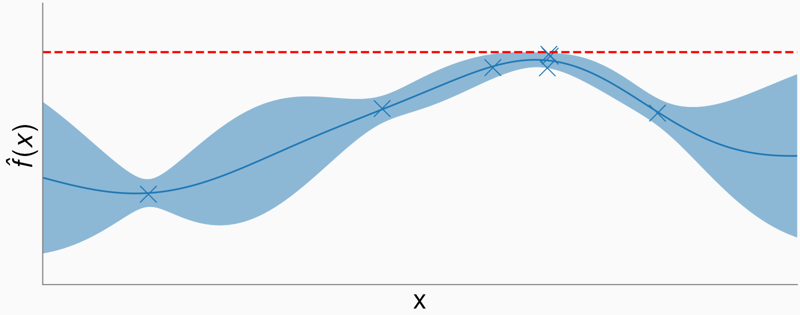
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

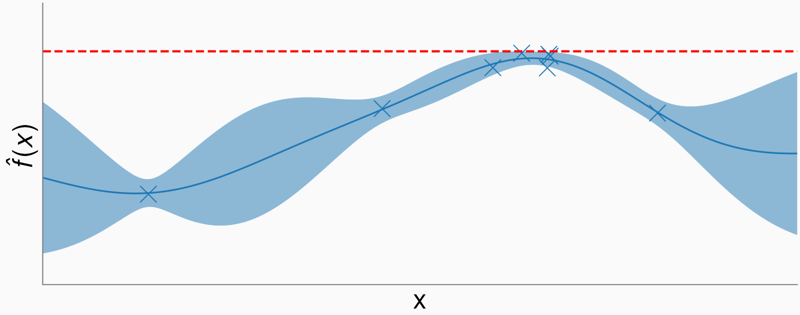
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

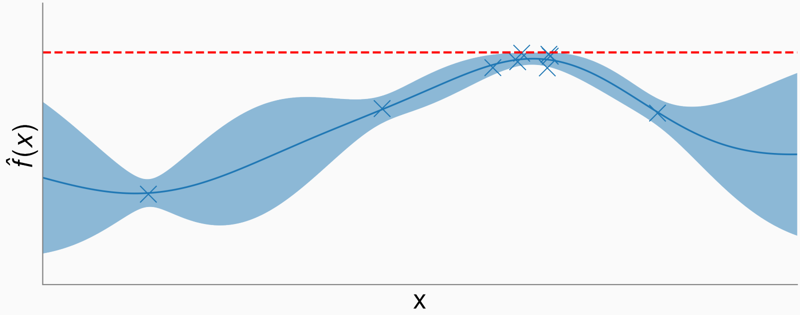
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

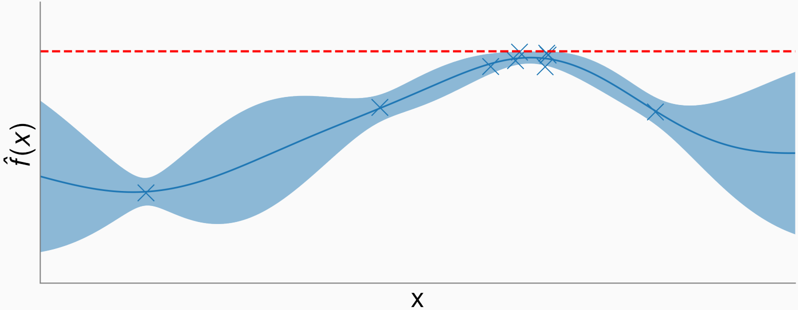
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize f.

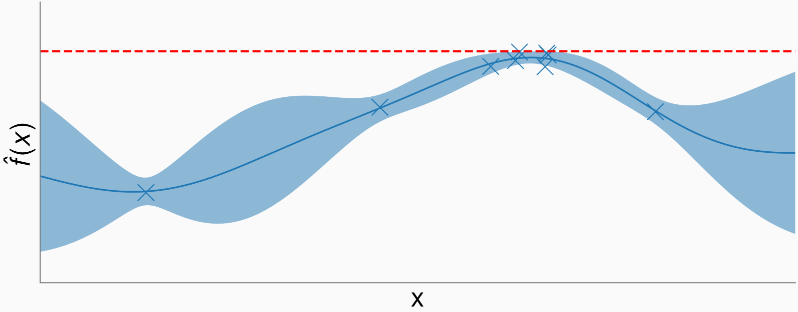
Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Convergence guarantee: $f(x_t) \longrightarrow f(x^*) \quad \text{as} \quad t \longrightarrow \infty$

Idea: Use confidence intervals to efficiently optimize f.

Example: GP-UCB (**G**aussian **P**rocess - **U**pper **C**onfidence **B**ound)



Convergence guarantee: $\frac{1}{T}\sum_{x=1}^{T} f(x^*) - f(x_t) \leq \mathcal{O}\left(1/\sqrt{T}\right)$

Extension 1: Safe Bayesian Optimization

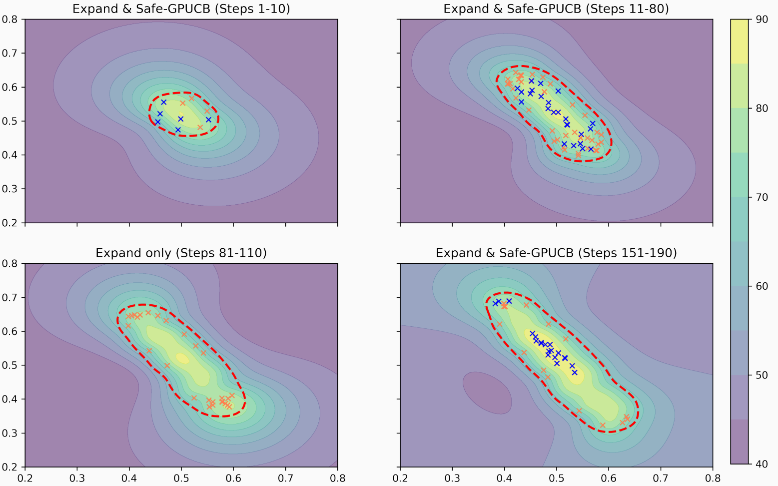
Objective: Keep a safety function $s(x)$ below a threshold c.

$$\max_{x \in \mathcal{X}} f(x) \quad \text{s.t.} \quad s(x) \leq c$$

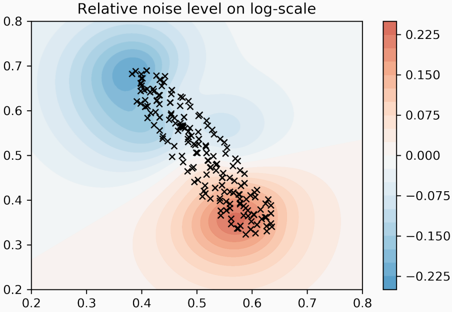
SafeOpt: [Sui et al.,(2015); Berkenkamp et al. (2016)]

Extension 1: Safe Bayesian Optimization

Safe Tuning of 2 Matching Quadrupoles at SwissFEL:

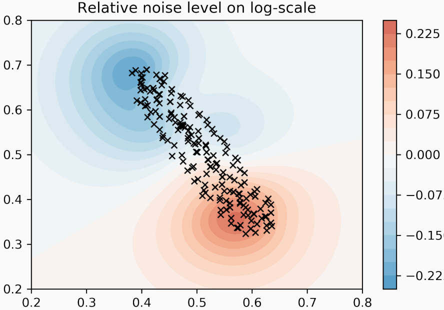


What if the noise variance depends on evaluation point?



Extension 2: Heteroscedastic Noise

What if the noise variance depends on evaluation point?



Standard approaches, like GP-UCB, are agnostic to noise level.

Information Directed Sampling: Bayesian optimization with heteroscedastic noise; including theoretical guarantees.
[Kirschner and Krause (2018); Russo and Van Roy (2014)]

Acknowledgments

Experiments at SwissFEL

Joined work with*Franziska Frei, Nicole Hiller, Rasmus Ischebeck, Andreas Krause, Morjmir Mutny*

Plots

Thanks to *Felix Berkenkamp* for sharing his python notebooks.

Pictures

Accelerator Structure: *Franziska Frei*

References

F. Berkenkamp, A. P. Schoellig, A. Krause., *Safe Controller Optimization for Quadrotors with Gaussian Processes*, ICRA, 2016

J. Kirschner and A. Krause, *Information Directed Sampling and Bandits with Heteroscedastic Noise*, ArXiv preprint, 2018

D. Russo and B. Van Roy, *Learning to Optimize via Information-Directed Sampling*, NIPS 2014

Y. Sui, A. Gotovos, J. W. Burdick, and A. Krause, *Safe exploration for optimization with Gaussian processes*, ICML 2015