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:

\( x = \text{Parameter settings on accelerator} \)

\( f(x) = \text{Pulse energy} \)

Goal:

\[ x^* = \arg \max_{x \in X} f(x) \]
Example:
\(x = \) Parameter settings on accelerator
\(f(x) = \) Pulse energy

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

\[
\ldots \text{ using only noisy evaluations } y_t = f(x_t) + \epsilon_t.
\]
Part 1)

A flexible & statistically sound model for $f$:
Gaussian Processes
Given: Measurements \((x_1, y_1), \ldots, (x_t, y_t)\).

Goal: Find statistical estimator \(\hat{f}(x)\) of \(f\).
Regularized linear least squares:

\[ \hat{\theta} = \arg \min_{\theta \in \mathbb{R}^d} \sum_{t=1}^{T} (x_t^\top \theta - y_t)^2 + \|\theta\|^2 \]
Least squares regression in a Hilbert space $\mathcal{H}$:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{t=1}^{T} (f(x_t) - y_t)^2 + \|f\|_{\mathcal{H}}^2$$
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 $\mathcal{H}$. 
\[ \hat{f} = \arg \min_{f \in \mathcal{H}} \sum_{t=1}^{T} (f(x_t) - y_t)^2 + \| f \|^2_{\mathcal{H}} \]
\[ \hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{t=1}^{T} (f(x_t) - y_t)^2 + \|f\|_{\mathcal{H}}^2 \]
\[ \hat{f} = \arg \min_{f \in \mathcal{H}} \sum_{t=1}^{T} (f(x_t) - y_t)^2 + \|f\|_{\mathcal{H}}^2 \]
**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.
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.
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.
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
**Idea:** Use confidence intervals to efficiently optimize $f$.

**Example:** Plausible Maximizers
**Idea:** Use confidence intervals to efficiently optimize $f$.

**Example:** Plausible Maximizers
Idea: Use confidence intervals to efficiently optimize $f$.

Example: GP-UCB (Gaussian Process - Upper Confidence Bound)
**Idea:** Use confidence intervals to efficiently optimize $f$.

**Example:** GP-UCB (Gaussian Process - Upper Confidence Bound)

\[
\hat{f}(x)
\]
Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize $f$.

Example: GP-UCB (Gaussian Process - Upper Confidence Bound)
Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize $f$.

Example: GP-UCB (Gaussian Process - Upper Confidence Bound)
Idea: Use confidence intervals to efficiently optimize \( f \).

Example: GP-UCB (Gaussian Process - Upper Confidence Bound)
Bayesian Optimization: GP-UCB

**Idea:** Use confidence intervals to efficiently optimize $f$.

**Example:** GP-UCB (Gaussian Process - Upper Confidence Bound)
Idea: Use confidence intervals to efficiently optimize $f$.

Example: GP-UCB (Gaussian Process - Upper Confidence Bound)
Idea: Use confidence intervals to efficiently optimize $f$.

Example: GP-UCB (Gaussian Process - Upper Confidence Bound)
**Bayesian Optimization: GP-UCB**

**Idea:** Use confidence intervals to efficiently optimize $f$.

**Example:** GP-UCB (Gaussian Process - Upper Confidence Bound)
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:** $f(x_t) \rightarrow f(x^*)$ as $t \rightarrow \infty$
Bayesian Optimization: GP-UCB

Idea: Use confidence intervals to efficiently optimize $f$.

Example: GP-UCB (Gaussian Process - Upper Confidence Bound)

Convergence guarantee: \[ \frac{1}{T} \sum_{x=1}^{T} f(x^*) - f(x_t) \leq O \left( \frac{1}{\sqrt{T}} \right) \]
Objective: Keep a safety function $s(x)$ below a threshold $c$.

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

SafeOpt: [Sui et al., (2015); Berkenkamp et al. (2016)]
Safe Tuning of 2 Matching Quadrupoles at SwissFEL:
What if the noise variance depends on evaluation point?

Relative noise level on log-scale
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

