Bayesian Optimization at LCLS
Using Gaussian Processes for Automated Tuning

Mitchell McIntire
mcintire@slac.stanford.edu
The FEL beam is focused and tuned using quadrupole magnets located along the beam line.

Tuning has historically been done by hand by machine operators and is very time consuming.

Our approach: treat FEL tuning as an optimization problem.
FEL Tuning at LCLS

• The FEL beam is focused and tuned using quadrupole magnets located along the beam line.
• Tuning has historically been done by hand by machine operators and is very time consuming.
• Our approach: treat FEL tuning as an optimization problem.
2015 Tuning Data

• Data mining from the 2015 quad settings allows us to collect data on tuning events.
• We filter for non-parasitic tuning events, removing e.g. events like:

LTU Quads
2015 Tuning Data

- Data mining from the 2015 quad settings allows us to collect data on tuning events.
- We filter for non-parasitic tuning events (below).
- These account for 200+ hours of machine time in 2015.
Outline

• Optimization
• Bayesian optimization
• Gaussian processes
• Integration into LCLS
• Preliminary results
Review: Optimization

- Optimization in brief: find $x$ that gives the best $y$
- Requires evaluation of the objective function
Review: Optimization

- Many types of optimization algorithms
  - Gradient methods
Review: Optimization

- Many types of optimization algorithms
  - Simulated annealing
Review: Optimization

- Many types of optimization algorithms
  - Simulated annealing

![Graph showing a function with a peak and a temperature scale T]
Review: Optimization

- Many types of optimization algorithms
  - Genetic algorithms

https://commons.wikimedia.org/wiki/File:St_5-xband-antenna.jpg
What do we want from an optimizer?

- Speed
- Capability of handling noise
- Interpretability
Review: Optimization

What do we want from an optimizer?

- Speed
- Capability of handling noise
- Interpretability

Bayesian optimization!
Bayesian optimization: High-level overview

Machine’s perspective:
Bayesian optimization: High-level overview

Machine’s perspective:

Optimizer

$y_{t-1}$
Bayesian optimization: High-level overview

Machine’s perspective:

Optimizer

\( y_{t-1} \)
Bayesian optimization: High-level overview

Machine’s perspective:

Optimizer

$x_t$
Bayesian optimization: High-level overview

Machine’s perspective:

Optimizer

$x_t$
Bayesian optimization: High-level overview

Machine’s perspective:
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

Model_{t-1}  Machine
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

\[ y_{t-1} \]

Model_{t-1} → Machine
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

\[(x_{t-1}, y_{t-1})\]

\[\text{Model}_{t-1}\]

\[\text{Machine}\]

\[y_{t-1}\]
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

\[(x_{t-1}, y_{t-1}) \rightarrow \text{Model}_{t-1} \rightarrow \text{Machine} \rightarrow y_{t-1}\]
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

Model\(_t\)  

Machine
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

Model_t

Machine

$\mathbf{x}_t, \hat{y}_t$

Acquisition
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

\[(x_t, \hat{y}_t)\]
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

Model_t

Machine

\((x_t, \hat{y}_t)\)

\(x_t\)
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

\( (x_t, \hat{y}_t) \)
Bayesian optimization: Mid-level overview

Optimizer’s perspective:

$\text{Model}_t \rightarrow (x_t, \hat{y}_t) \rightarrow \text{Machine} \rightarrow y_t$

$y_t$ is the output of the machine, $\hat{y}_t$ is the model's prediction for $x_t$, and $x_t$ is the input to the model.
The optimizer is fully defined by its:

- Model
- Acquisition function
Model: High-level

- The model should be a mapping from parameter space into probability distributions.
- For us, the model gives a probability distribution over pulse energies for each possible set of quad strengths.
A Gaussian process (GP) is a nonparametric model that predicts a normal distribution at each point.

The shape of the distribution is calculated by inference over training data:

$$P(y_{\text{new}} | x_{\text{new}}, X_{\text{train}}, Y_{\text{train}}) \sim \mathcal{N}(\mu(x), \sigma(x))$$
GP Covariance Functions

A GP is fully defined by its:

- Prior mean function
- Covariance function
- Training data

A covariance function is essentially a similarity measure between points in parameter space: $K: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
GP Covariance Functions

- Covariance functions encode the type of functions we think are likely, i.e. the way the data ‘should look’.
- A common choice is the squared exponential kernel:

\[
K(x_1, x_2) = \theta e^{-\frac{(x_1 - x_2)^T(x_1 - x_2)}{2\ell^2}}
\]

http://pythonhosted.org/infpy/gps.html
GP Covariance Functions

- Covariance functions encode the type of functions we think are likely, i.e. the way the data ‘should look’.
- We use a slight variation:

\[
K(x_1, x_2) = \theta e^{-\frac{(x_1 - x_2)^T(x_1 - x_2)}{2\ell^2}}
\]

\[
K(x_1, x_2) = \theta e^{-(x_1 - x_2)^T \Lambda^{-1}(x_1 - x_2)}
\]
Other Covariance Functions

Covariance functions can encode complex beliefs about the type of functions that are likely:

http://pythonhosted.org/infpy/gps.html
Acquisition function

- Acquisition functions define exploration behavior.
- We use expected improvement (EI):

\[
I(x) = \max(f(x) - y^*, 0),
\]

\[
EI(x) = E[I(x)] = \int_{y^*}^{\infty} (y - y^*)P(y|x)dy
\]
Acquisition functions define exploration behavior.

We use expected improvement (EI):

\[
I(x) = \max(f(x) - y^*, 0),
\]

\[
\text{EI}(x) = E[I(x)] = \int_{y^*}^{\infty} (y - y^*)P(y|x)dy
\]

Easy to compute for a Gaussian distribution.
Example: Bayesian optimization
Example: Bayesian optimization

Acquisition point
Example: Bayesian optimization
Example: Bayesian optimization
Example: Bayesian optimization
Example: Bayesian optimization

![Diagram showing Acquisition point]
Example: Bayesian optimization
Example: Bayesian optimization
Example: Bayesian optimization

Acquisition point
Example: Bayesian optimization

Acquisition point
Example: Bayesian optimization
Integration into LCLS

- We use the following covariance function:

\[ K(x_1, x_2) = \theta e^{-(x_1 - x_2)^T \Lambda^{-1} (x_1 - x_2)} \]

- The hyperparameters \( \theta \) and \( \Lambda \) are calculated from historical data, e.g. historical deviation of a certain quad’s settings.

- Via the Ocelot GUI (right), arbitrary sets of quads can be selected and optimized.
• The GP’s full model can be observed for analysis and debugging, along with the acquisition function:
Analysis

• The GP’s full model can be observed for analysis and debugging, along with the acquisition function:
Results

- Early results are promising, and the optimizer can perform well optimizing 12 quads at once:

- Man vs. Machine tests pending…
Future Work

- We are not incorporating any physics into the optimizer!
- We want to use physical models and our knowledge about the system to augment the optimization procedure.
- One possibility is marginalizing over hidden variables:

\[ P(y|x, X, Y) = \int P(y|x, X, Y, z)P(z|X, Y)dz \]

- We could also work directly with physical parameters, e.g. \( \alpha \) and \( \beta \), which describe the shape of the beam.