Background: Multi-Armed Bandits

- **Problem:** Which machine has highest rate of payout?
- **Trade-off:** Exploration (trying a new machine) vs Exploitation (playing machine with best returns so far)
- **Regret:** Difference between reward of action, and reward of optimal action (with benefit of hindsight)
Background: Multi-Armed Bandits

• **Problem:** Which machine has highest rate of payout?

• **Trade-off:** Exploration (trying a new machine) vs Exploitation (playing machine with best returns so far)

• **Regret:** Difference between reward of action, and reward of optimal action (with benefit of hindsight)
Bayesian Optimization

Goal: Optimize unknown cost function (continuous version of bandit problem)
Bayesian Optimization

Goal: Optimize unknown cost function (continuous version of bandit problem)
Bayesian Optimization

Goal: Optimize unknown cost function (continuous version of bandit problem)
Bayesian Optimization

Goal: Optimize unknown cost function (continuous version of bandit problem)
Bayesian Optimization

The Bayesian Optimization Idea

Where should we evaluate next in order to improve the most?

Goal: Optimize unknown cost function (continuous version of bandit problem)
Bayesian Optimization

Problem: Which point should we evaluate next?
Bayesian Optimization

Problem: Which point should we evaluate next?
Bayesian Optimization

Idea 1: Model uncertainty about objective function

Idea 1: Model uncertainty about objective function
Bayesian Optimization

Idea 2: Define acquisition function that balances exploration and exploitation
Bayesian Optimization

pred var | pred mean | truth | evaluations
Bayesian Optimization
Bayesian Optimization
Bayesian Optimization
Intuition: Why does Bayes Opt work?

Idea: Use confidence bounds to adaptively eliminate regions in search space that are not likely to contain optimum.
Modeling Uncertainty
Predictive Posterior over Functions

\[ x^* = \arg\min_x f(x) \quad f \sim p(f \mid y) \]
Recap: Gaussian Processes

**Formal View:** Prior over Functions

\[ y_n \mid f \sim \text{Norm}(f(x_n), \sigma) \quad f \sim \text{GP}(\mu(x), k(x, x')) \]

**Practical View:** Generalization of Multivariate Normal

\[
\begin{bmatrix}
  y \\
  f^*
\end{bmatrix}
\sim \text{Norm}
\left( \begin{bmatrix}
  \mu(X) \\
  \mu(x^*)
\end{bmatrix},
\begin{bmatrix}
  k(X, X) + \sigma^2 I & k(X, x^*) \\
  k(x^*, X) & k(x^*, x^*)
\end{bmatrix}\right)
\]
Recap: Gaussian Processes

\[
\begin{bmatrix} y \\ f^* \end{bmatrix} \sim \text{Norm}
\left(
\begin{bmatrix} \mu(X) \\ \mu(x^*) \end{bmatrix},
\begin{bmatrix} k(X,X) + \sigma^2 I & k(X,x^*) \\ k(x^*,X) & k(x^*,x^*) \end{bmatrix}
\right)
\]

**Predictive Posterior:** Distribution on \( f^* \) for a new point \( x^* \)

\[
f^* \mid y \sim \text{Norm}(f ; \tilde{\mu}(x^*), \tilde{k}(x^*,x^*)),
\]

\[
\tilde{\mu}(x^*) = \mu(x^*) + k(x^*,X) \left( k(X,X) + \sigma^2 I \right)^{-1} (y - \mu(X)),
\]

\[
\tilde{k}(x^*,x^*) = k(x^*,x^*) - k(x^*,X) \left( k(X,X) + \sigma^2 I \right)^{-1} k(X,x^*).
\]
Recap: Gaussian Processes

\[
\begin{bmatrix}
  y \\
  f^* 
\end{bmatrix}
\sim \text{Norm}
\left( \begin{bmatrix}
  \mu(X) \\
  \mu(x^*) 
\end{bmatrix}, \begin{bmatrix}
  k(X, X) + \sigma^2 I & k(X, x^*) \\
  k(x^*, X) & k(x^*, x^*) 
\end{bmatrix} \right)
\]

**Predictive Posterior:** Distribution on \( f^* \) for a new point \( x^* \)

\[
f^* \mid y \sim \text{Norm}(f; \tilde{\mu}(x^*), \tilde{\sigma}(x^*))
\]

\[
\tilde{\mu}(x^*) = \mu(x^*) + k(x^*, X) \left( k(X, X) + \sigma^2 I \right)^{-1} (y - \mu(X)),
\]

\[
\tilde{\sigma}(x^*) = k(x^*, x^*) - k(x^*, X) \left( k(X, X) + \sigma^2 I \right)^{-1} k(X, x^*).\]
Bayesian Optimization with GPs

\[ x^* = \arg \min_x f(x) \quad f(x) \mid y \sim \text{Norm}(\tilde{\mu}(x), \tilde{\sigma}(x)), \]

\[ \tilde{\mu}(x^*) = \mu(x^*) + k(x^*, X) \left( k(X, X) + \sigma^2 I \right)^{-1} (y - \mu(X)), \]

\[ \tilde{\sigma}(x^*) = k(x^*, x^*) - k(x^*, X) \left( k(X, X) + \sigma^2 I \right)^{-1} k(X, x^*). \]
Choices of Kernel Functions

**Squared-Exponential**

\[ C(x, x') = \alpha \exp \left\{ -\frac{1}{2} \sum_{d=1}^{D} \left( \frac{x_d - x'_d}{\ell_d} \right)^2 \right\} \]

**Matérn**

\[ C(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu} r}{\ell} \right)^\nu K_\nu \frac{\sqrt{2\nu} r}{\ell} \]

**“Neural Network”**

\[ C(x, x') = \frac{2}{\pi} \sin^{-1} \left\{ \frac{2x^T \Sigma x'}{\sqrt{(1 + 2x^T \Sigma x)(1 + 2x'^T \Sigma x')}} \right\} \]

**Periodic**

\[ C(x, x') = \exp \left\{ -\frac{2 \sin^2 \left( \frac{1}{2} (x - x') \right)}{\ell^2} \right\} \]
Acquisition Functions
Acquisition Functions

\[ x^* = \underset{x}{\text{argmin}} f(x) \quad f(x) | y \sim \text{Norm}(\tilde{\mu}(x), \tilde{\sigma}(x)), \]

1. **Exploration**: Evaluate *highest* posterior uncertainty
2. **Exploitation**: Evaluate *lowest* posterior mean
Acquisition Functions

\[ x^* = \arg\min_x f(x) \quad f(x) | y \sim \text{Norm}(\tilde{\mu}(x), \tilde{\sigma}(x)), \]

Exploration-exploitation trade-off: \[ \tilde{\mu}(x) - \kappa \tilde{\sigma}(x) \]
Upper/Lower Confidence Bounds

$$\text{UCB}(x) = \tilde{\mu}(x) + \kappa \tilde{\sigma}(x) \quad \text{LCB}(x) = \tilde{\mu}(x) - \kappa \tilde{\sigma}(x)$$

Can derive sequence of values $\kappa_n$ with provably optimal regret bounds, but tuning often needed in practice.
Probability of Improvement

\[ P(f(x^*) \leq \mu^-) = \Phi \left( \frac{\mu^- - \bar{\mu}(x)}{\bar{\sigma}(x)} \right) \quad \mu^- = \arg\min_n \bar{\mu}(x_n) \]

Not used often, but works well when optimal value is known.
Expected Improvement

\[ EI(x) = \mathbb{E}_{p(f|y)} \left[ \max \{0, \mu^- - f(x)\} \right] \]

\[ = (\mu^- - \widetilde{\mu}(x)) \Phi(Z) + \widetilde{\sigma}(x) N(Z; 0, 1) \]

\[ Z = \frac{\mu^- - \widetilde{\mu}(x)}{\widetilde{\sigma}(x)} \]

\[ \mu^- = \arg\min_n \widetilde{\mu}(x_n) \]
**Entropy Search**

**Idea:** model uncertainty about location of optimum $p(x^* | y_{1:n})$

$$H[x^* | y_{1:n}] - \mathbb{E}_{p(y^* | y_{1:n})}[H[x^* | y_{1:n}, y_{n+1}]]$$
Choosing Kernel Hyperparameters
Choices for GP Hyperparameters

\[ C(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu}}{\ell} \right)^\nu K_\nu \frac{\sqrt{2\nu}}{\ell} \]

\[ \nu = 1/2 \]

\[ \nu = 3/2 \]

\[ \nu = 5/2 \]

\[ \nu = \infty \]

Matérn: \( \nu \) determines how many times differentiable
MCMC for GP Hyperparameters

Idea: define prior over kernel parameter, a GP likelihood and take expectation of acquisition function over posterior.

\[ \hat{a}(x) = \int a(x ; \theta) p(\theta \mid \{x_n, y_n\}_{n=1}^N) \, d\theta \]

\[ \approx \frac{1}{K} \sum_{k=1}^{K} a(x ; \theta^{(k)}) \quad \theta^{(k)} \sim p(\theta \mid \{x_n, y_n\}_{n=1}^N) \]
MCMC for GP Hyperparameters

Posterior samples for 3 different length scales

Expected improvement for each length scale

Integrated expected improvement
MCMC for GP Hyperparameters

Optimizing SGD & Regularization Parameters for Logistic Regression

[Snoek, Larochelle & Adams, NIPS 2012]
Use GP to model Computational Cost

Tuning Hyperparameters for Deep Convolutional Neural Nets (CIFAR10)

Classification Error vs. Time (Hours)

- GP EI
- MCMC
- GP EI per Second
- State-of-the-art

[Snoek, Larochelle & Adams, NIPS 2012]
Open Source Implementations

https://github.com/HIPS/Spearmint
Open Source Implementations

https://github.com/probprog/bopp

[Bopp: Bayesian Optimization for Probabilistic Programs](https://github.com/probprog/bopp)

[Rainforth, Le, van de Meent, Osborne, Wood, NIPS 2016]