Bayesian Optimization

A method for finding a maximum for expensive cost functions.

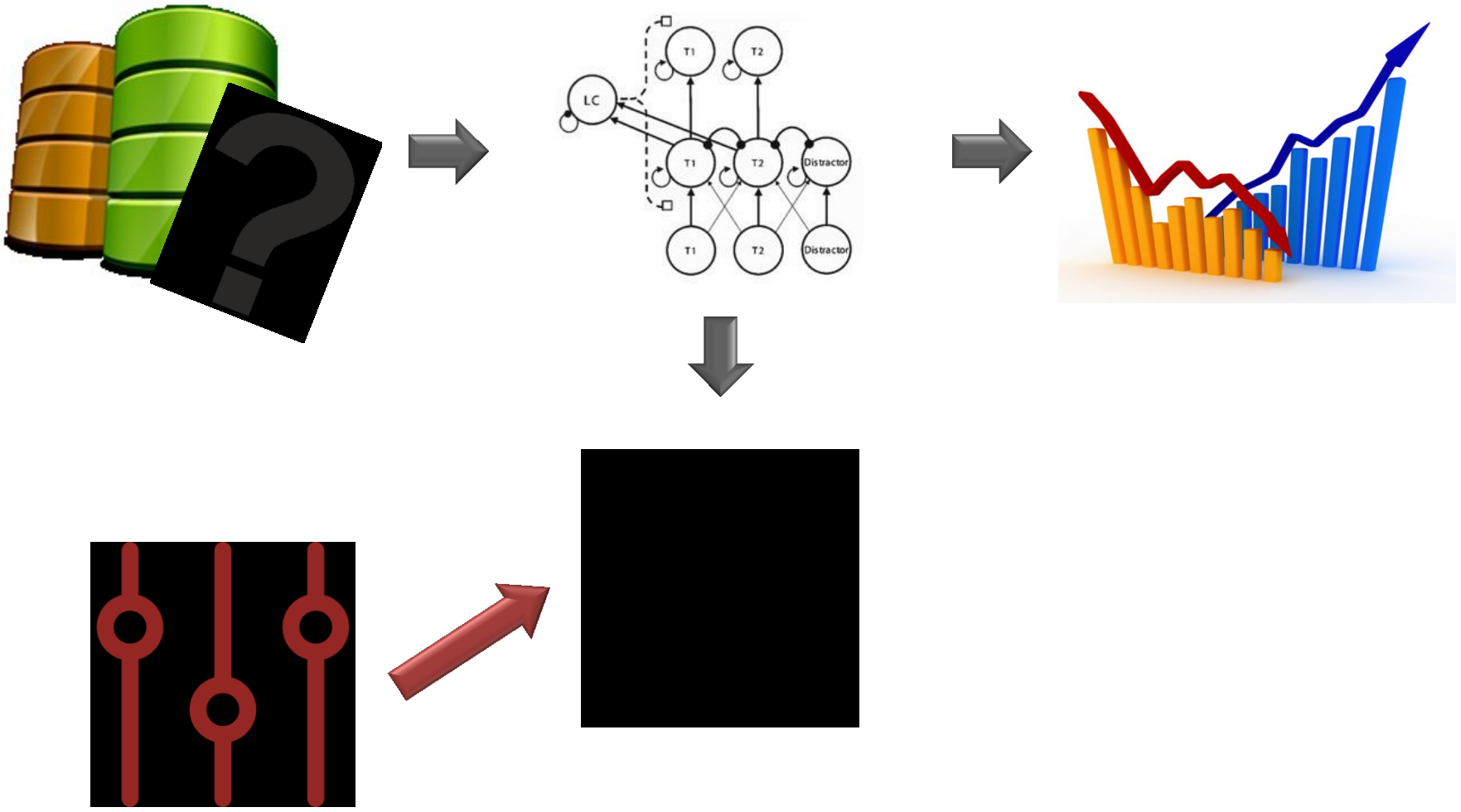
Outline

1. Example
2. Bayesian Optimization Overview
3. More Example Applications
4. Bayesian Optimization Properties, Algorithm, Components
    - Component 1: Gaussian Processes
    - Component 2: Acquisition Functions

Outline

1. Example
2. Bayesian Optimization Overview
3. More Example Applications
4. Bayesian Optimization Properties, Algorithm, Components
    - Component 1: Gaussian Processes
    - Component 2: Acquisition Functions

Machine Learning Pipeline



Hyperparameters

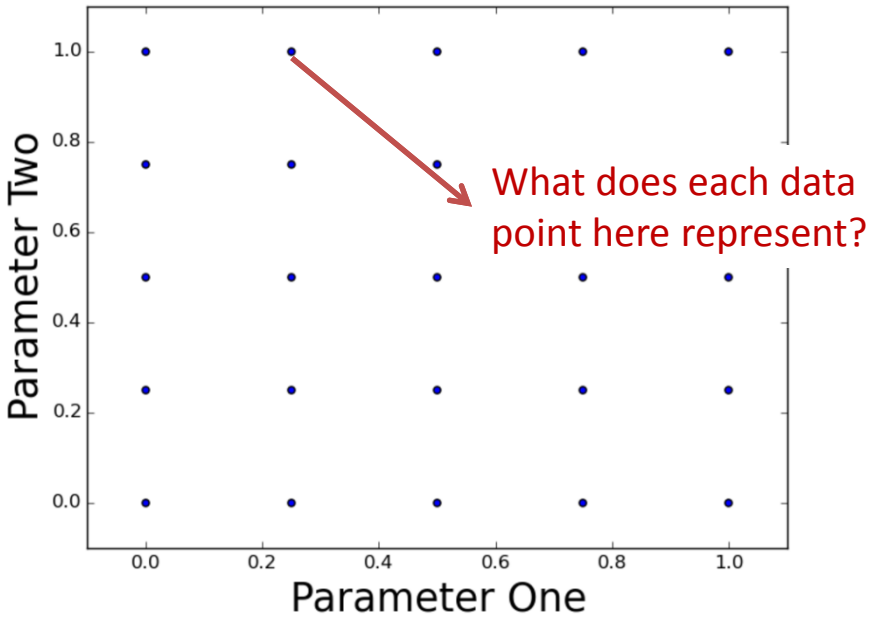
- The magic numbers!
- Use cross validation to measure parameter quality

$$x^* = arg\max_x f(x)$$

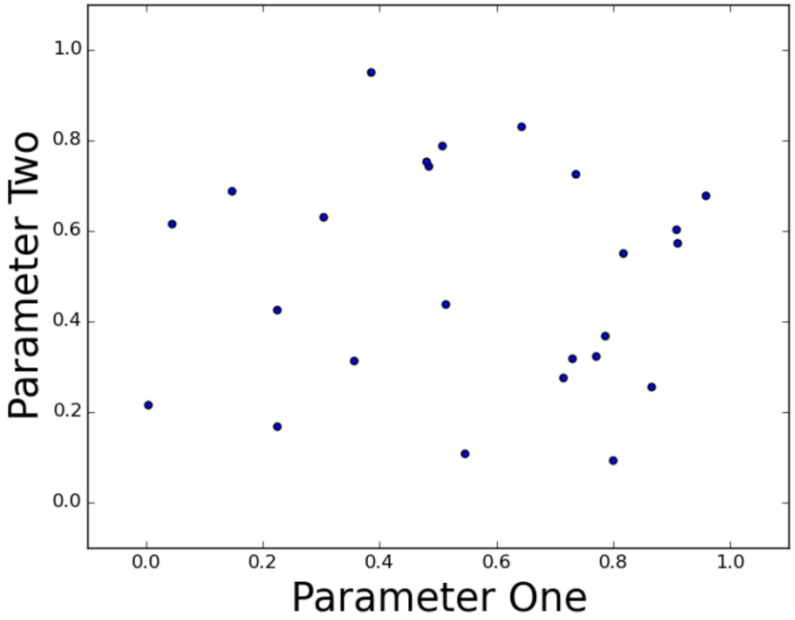
Cross Validation

How to choose values to test?

Grid search



Random search

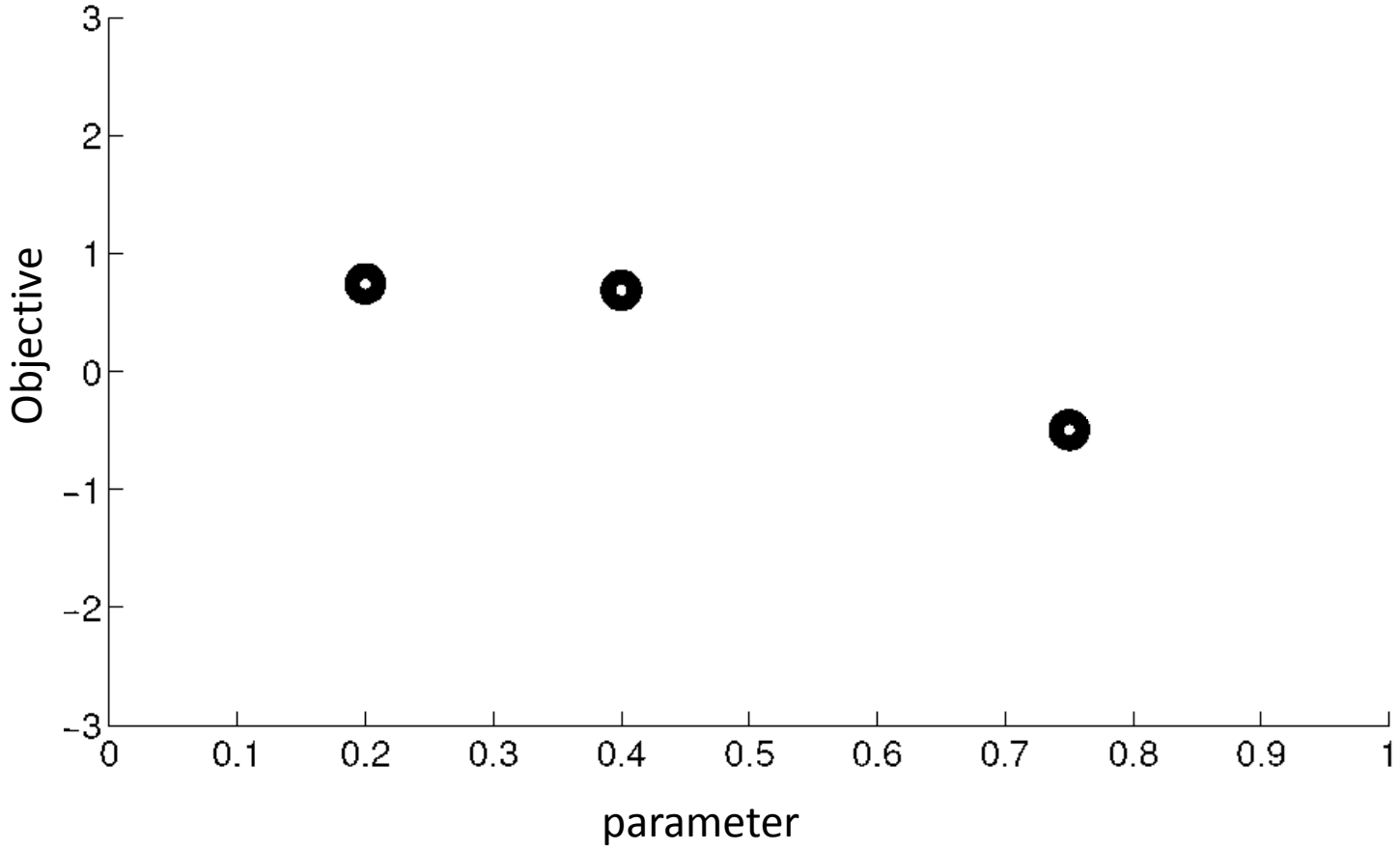


Simple Implementation

Scales poorly with number of parameters.
Many expensive training cycles required

Can We Do Better?
Bayesian Optimization

- Build a probabilistic model for the objective

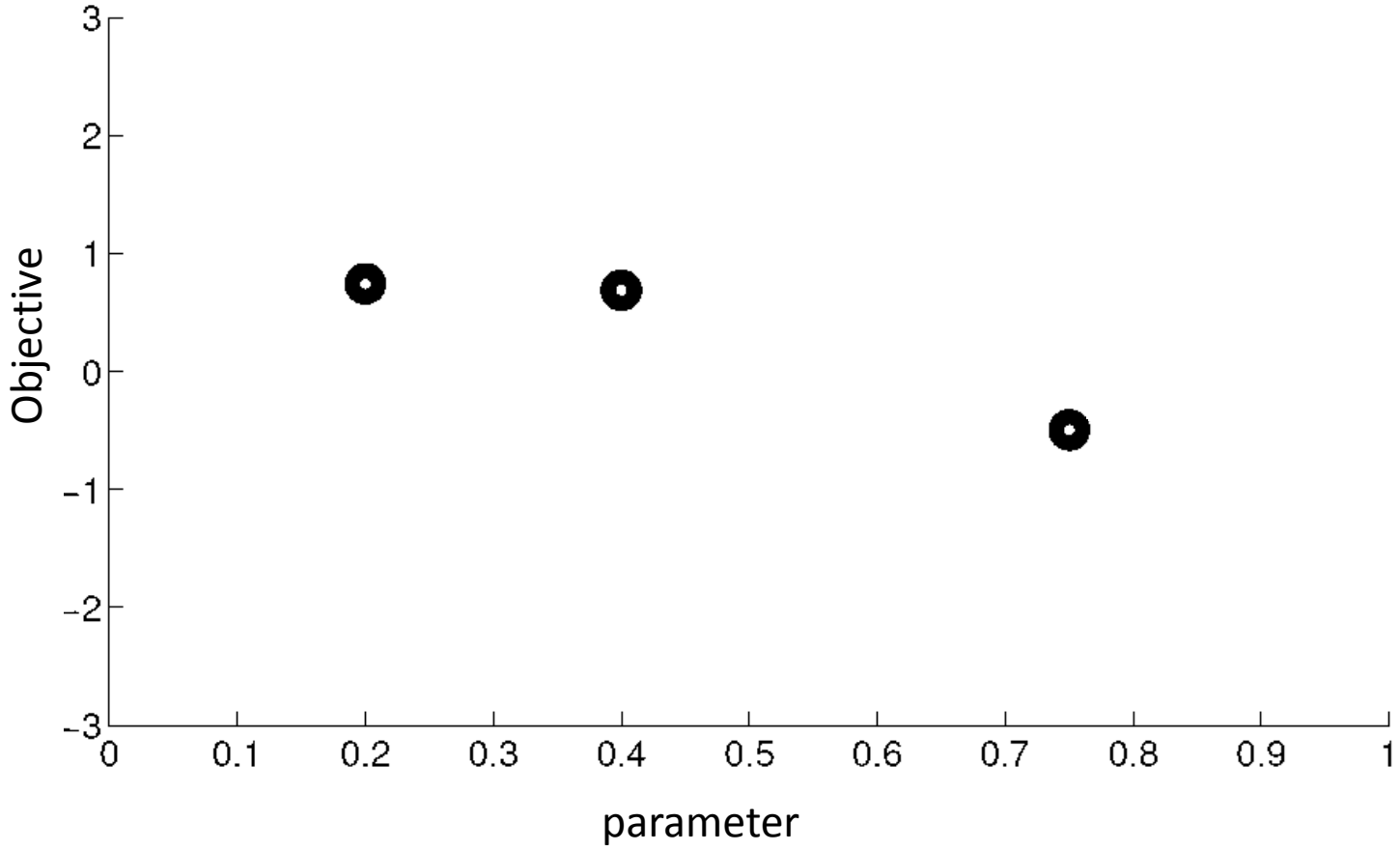


Outline

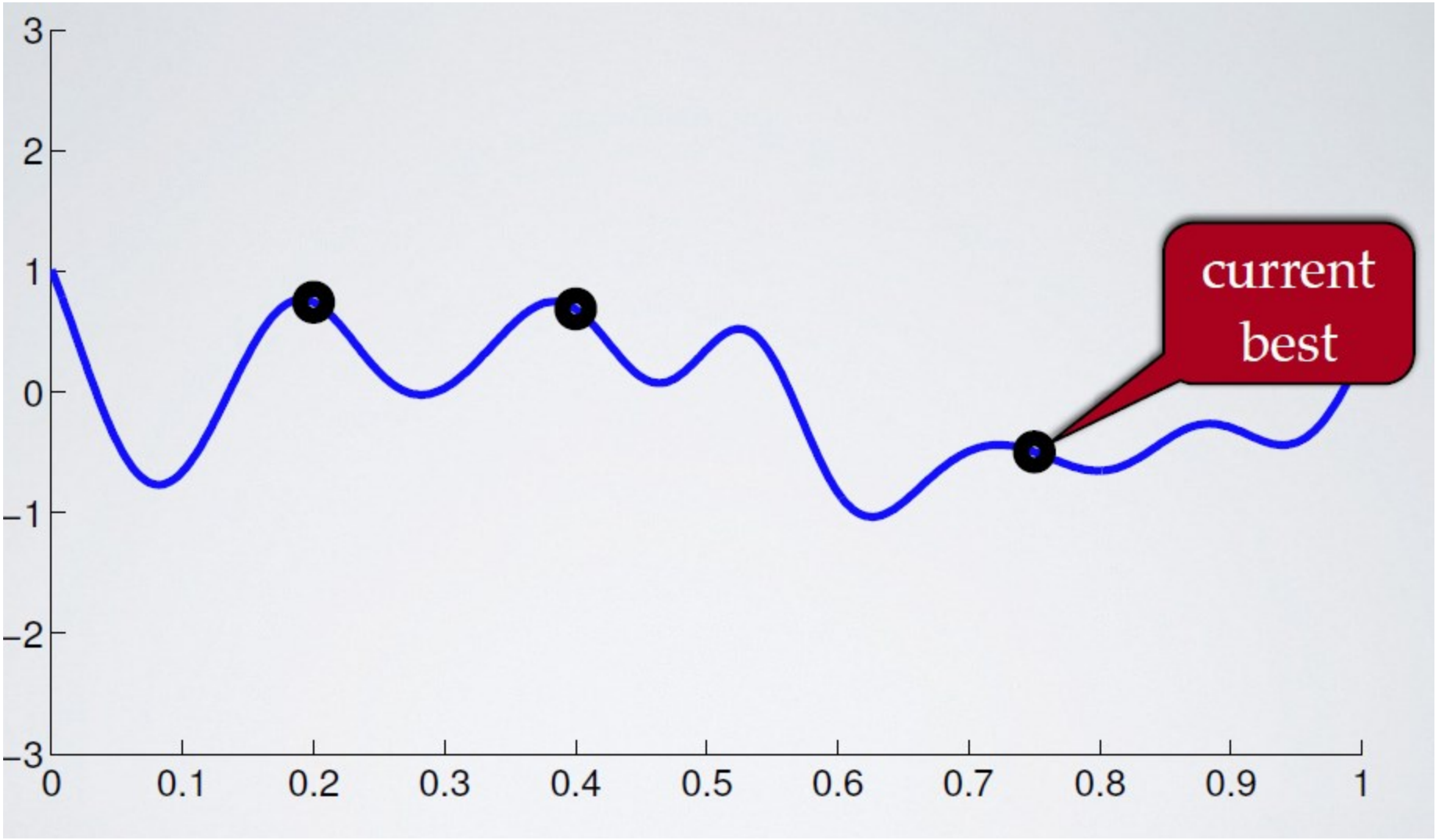
1. Example
2. Bayesian Optimization Overview
3. More Example Applications
4. Bayesian Optimization Properties, Algorithm, Components
    - Component 1: Gaussian Processes
    - Component 2: Acquisition Functions

Can We Do Better?
Bayesian Optimization

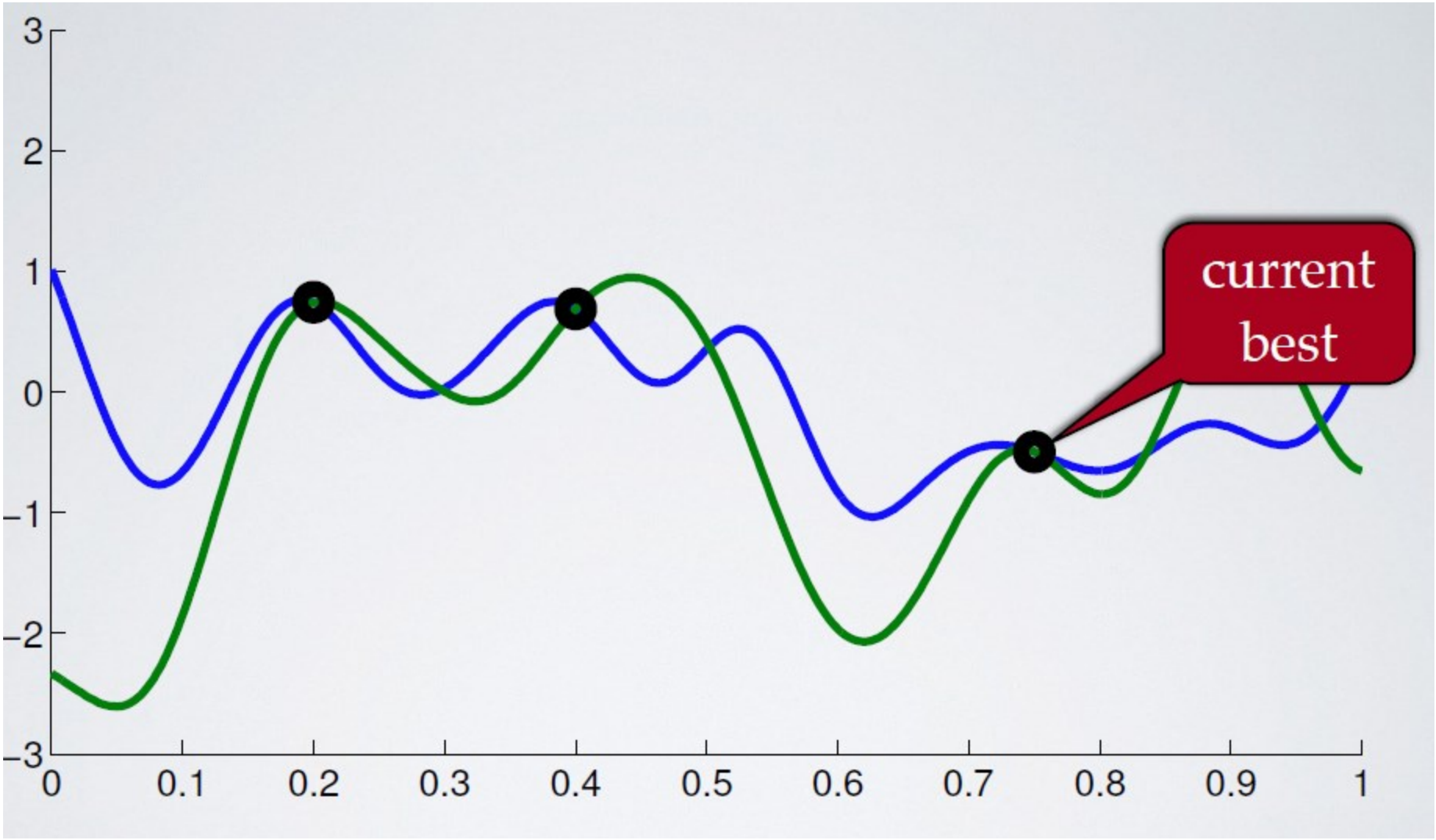
- Build a probabilistic model for the objective



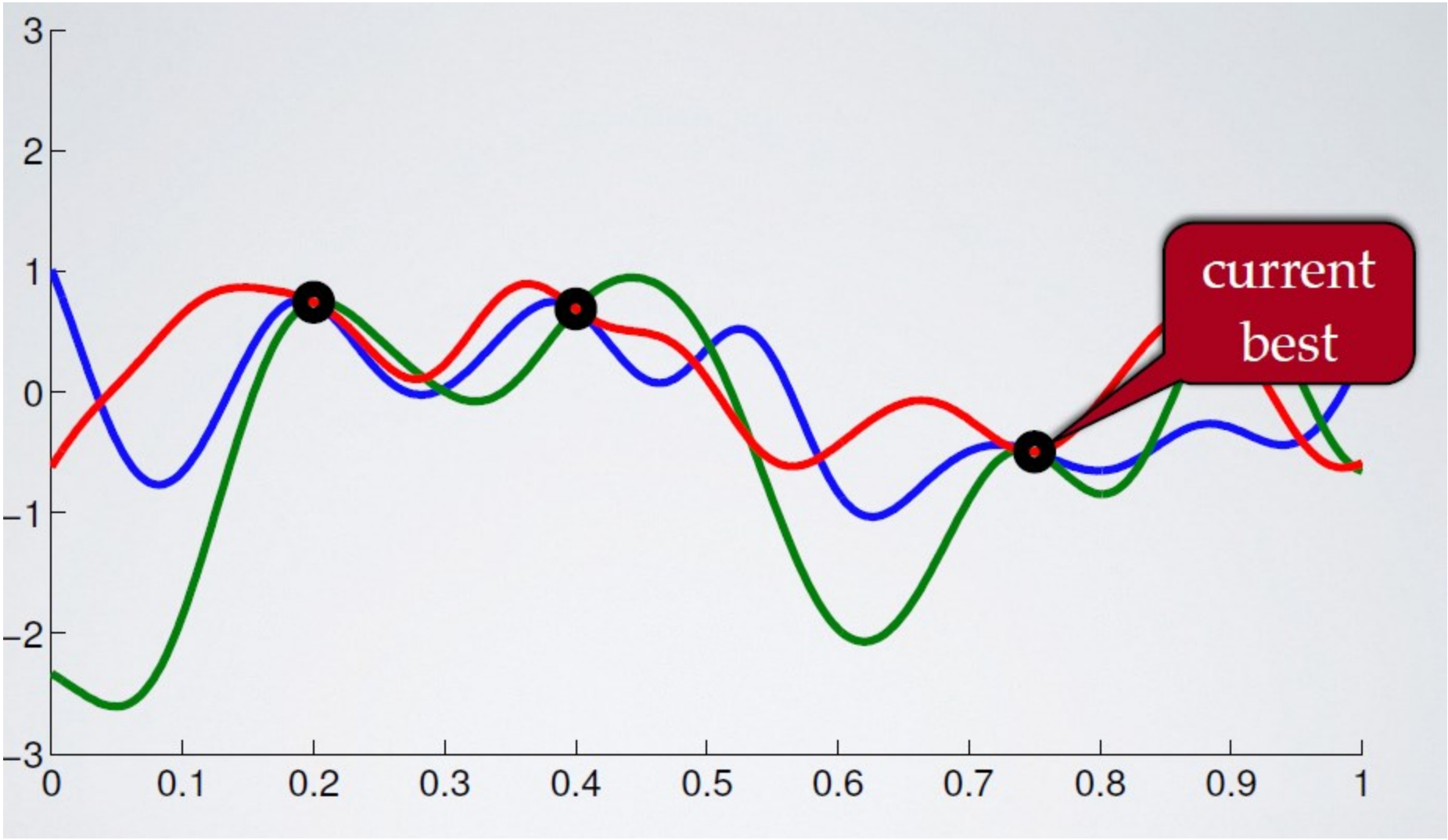
Bayesian Optimization



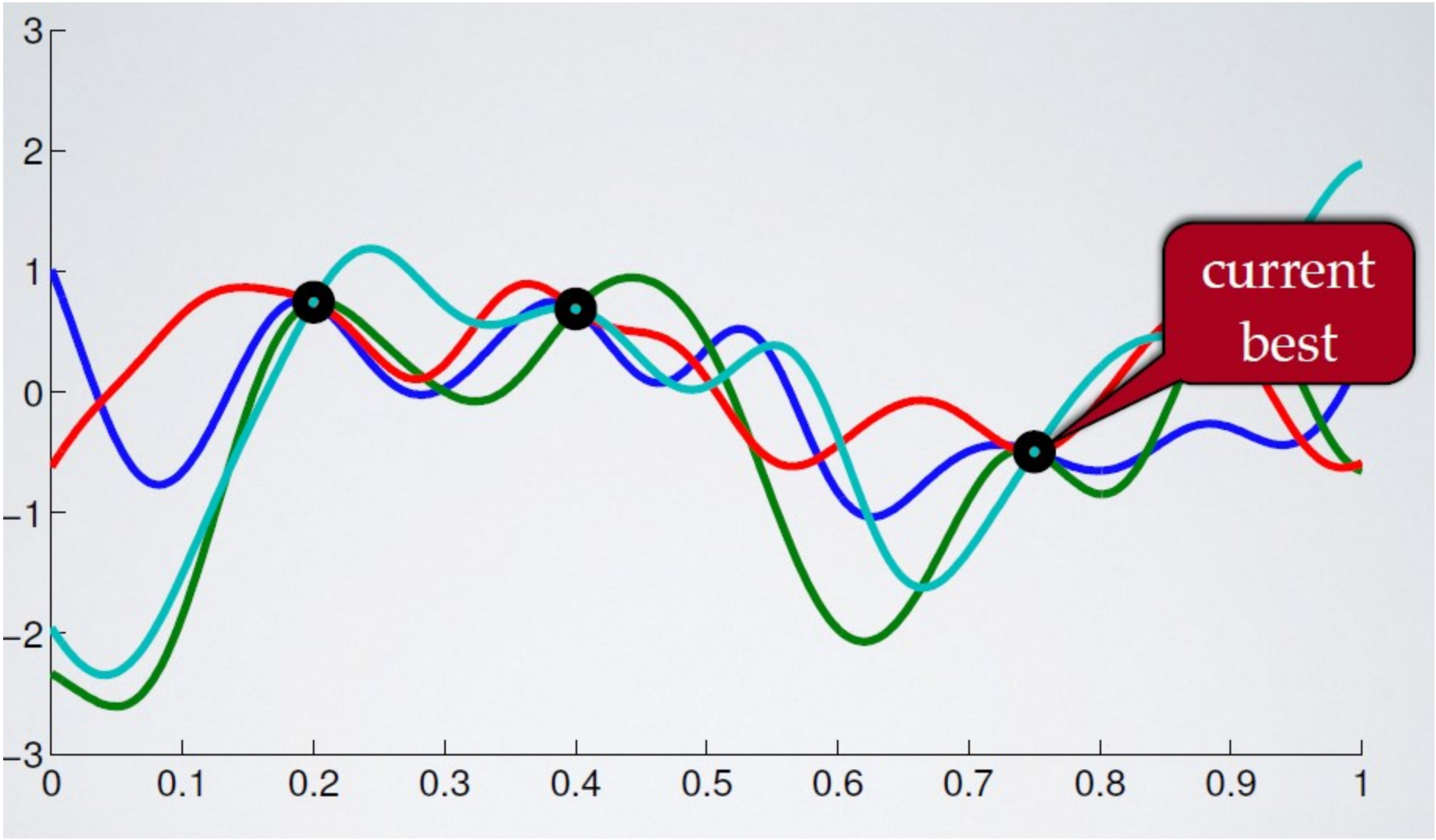
Bayesian Optimization



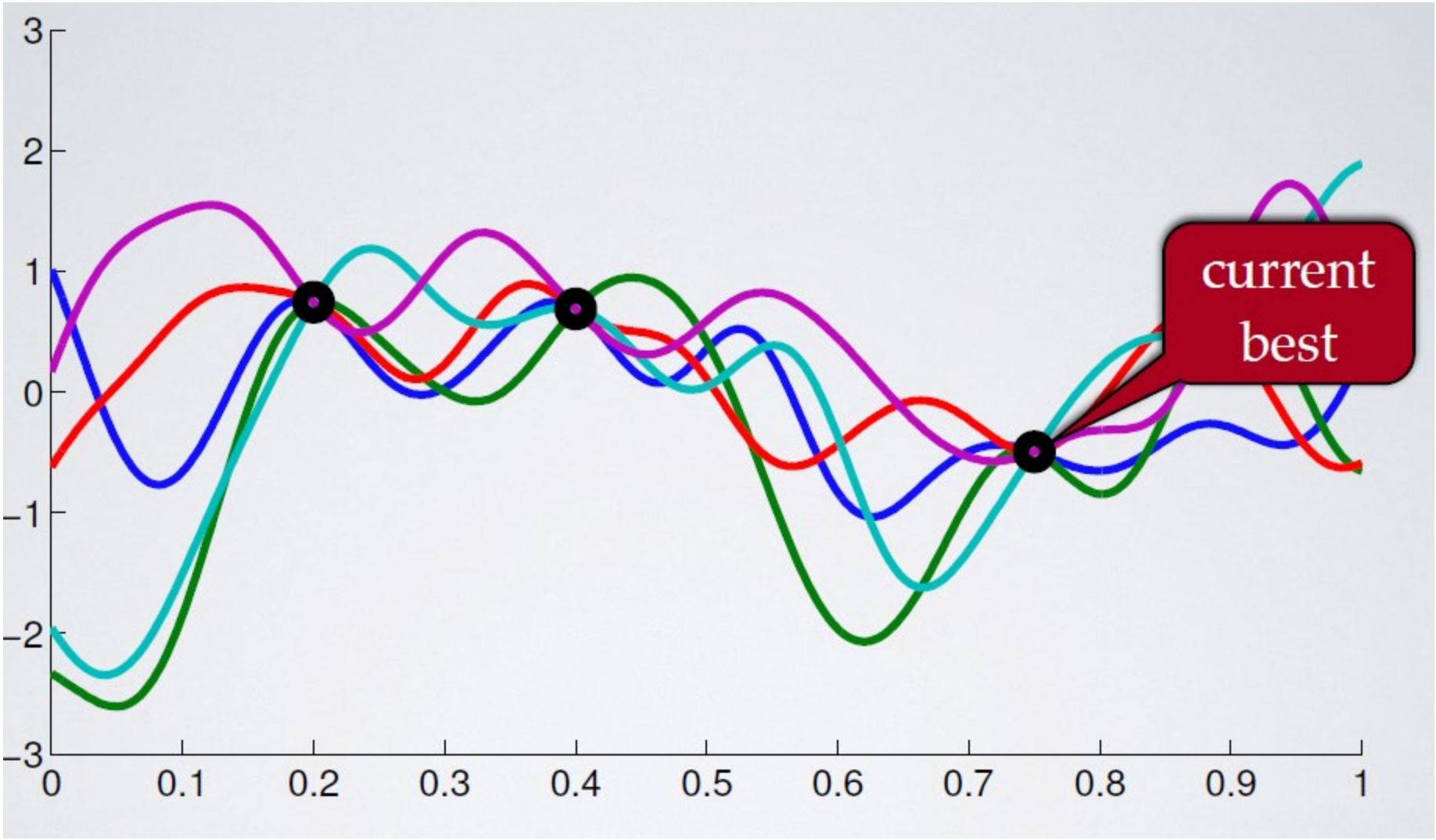
Bayesian Optimization



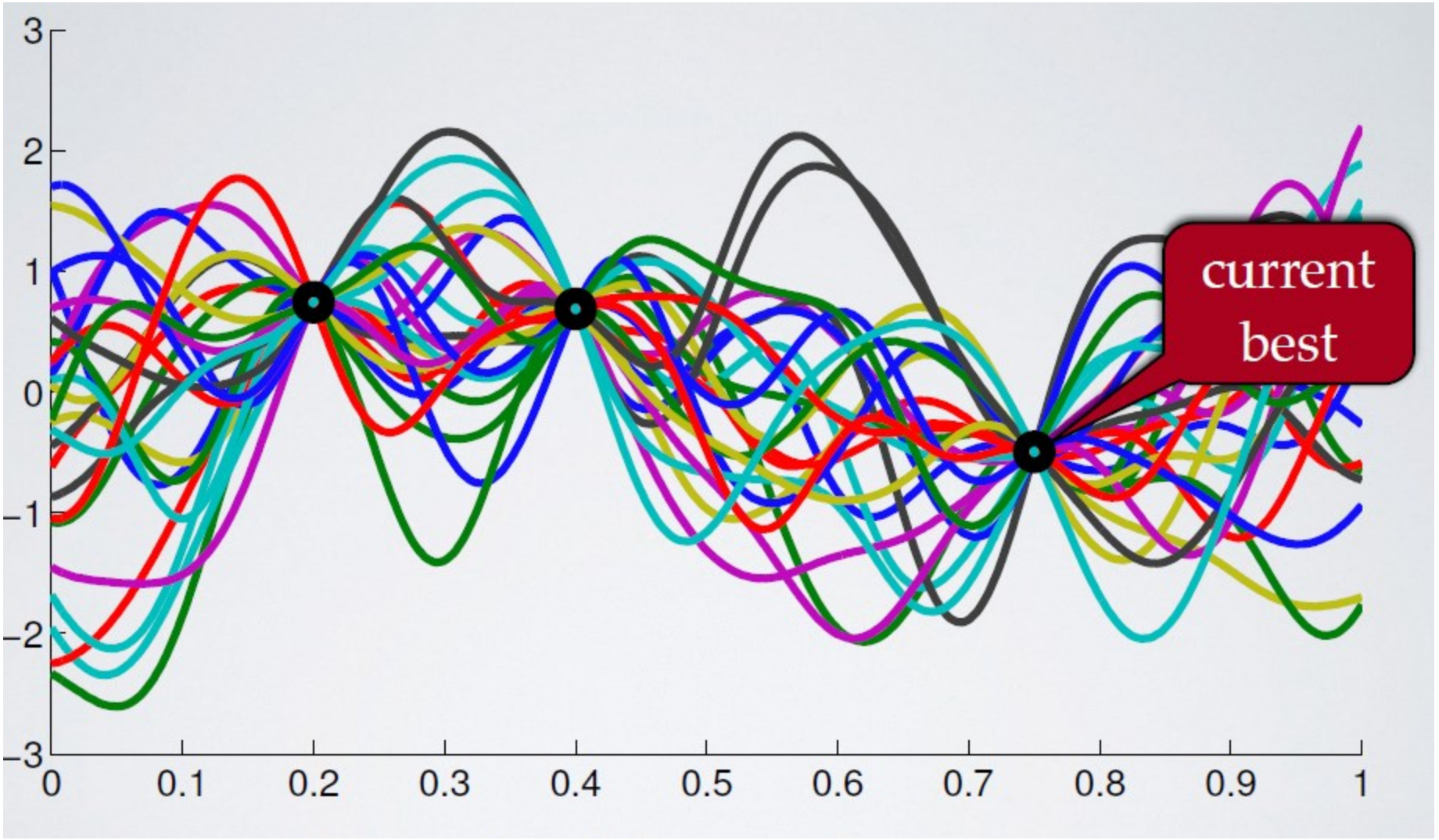
Bayesian Optimization



Bayesian Optimization

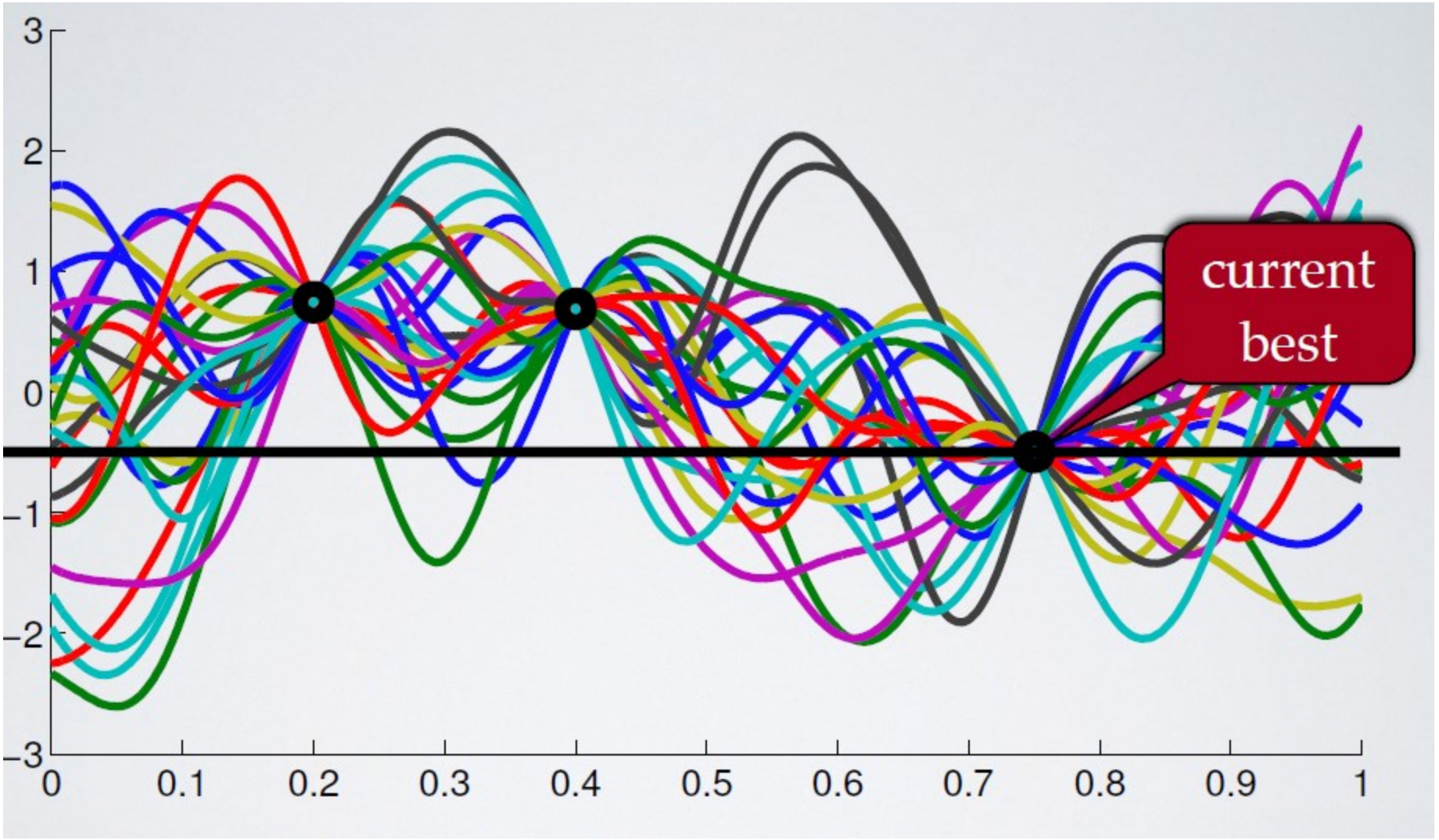


Bayesian Optimization



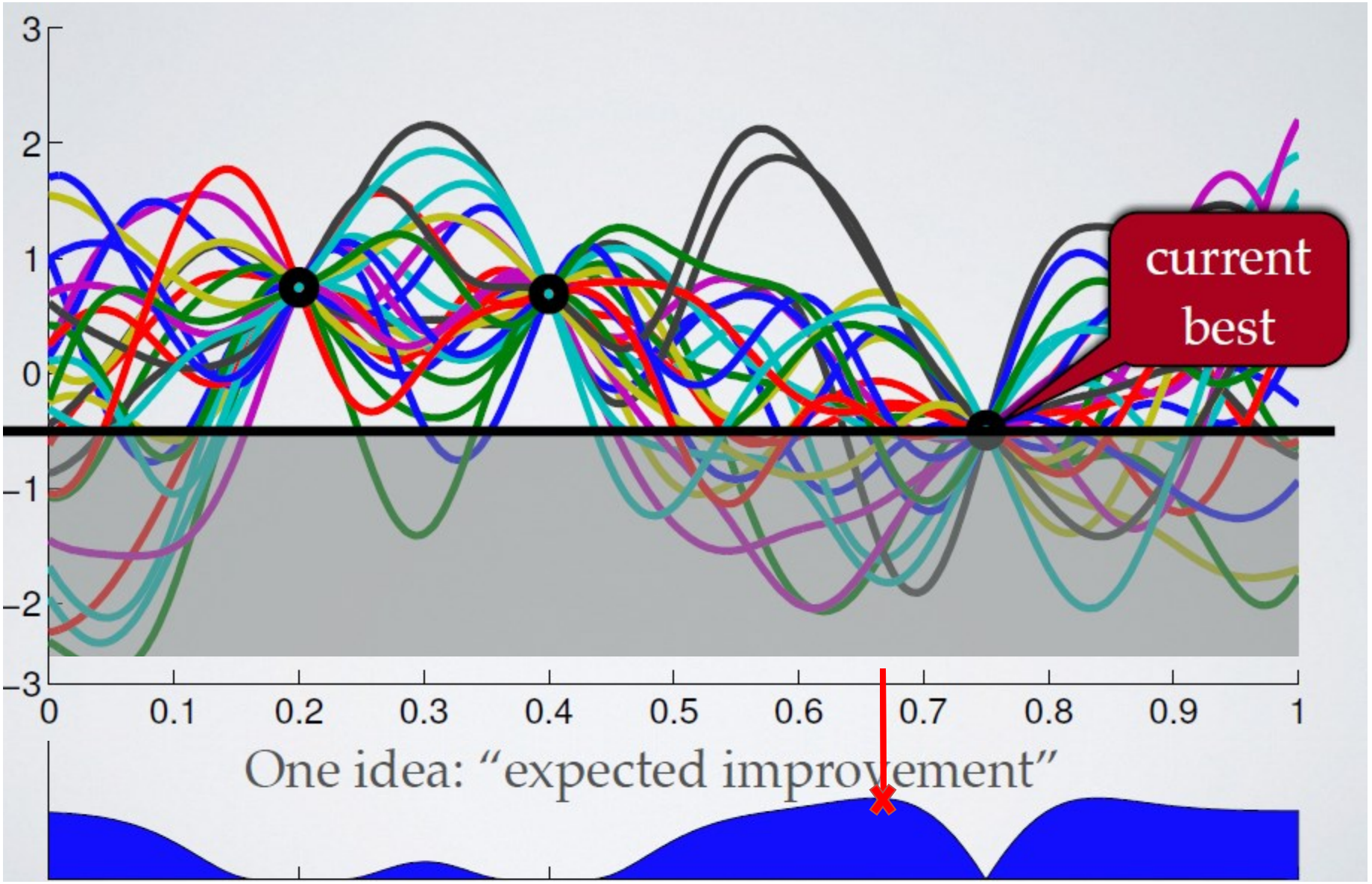
Where should we evaluate next in order to improve the most?

Bayesian Optimization

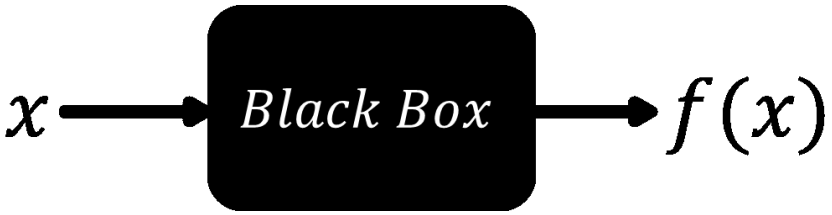


Where should we evaluate next in order to improve the most?

Bayesian Optimization



Bayesian Optimization

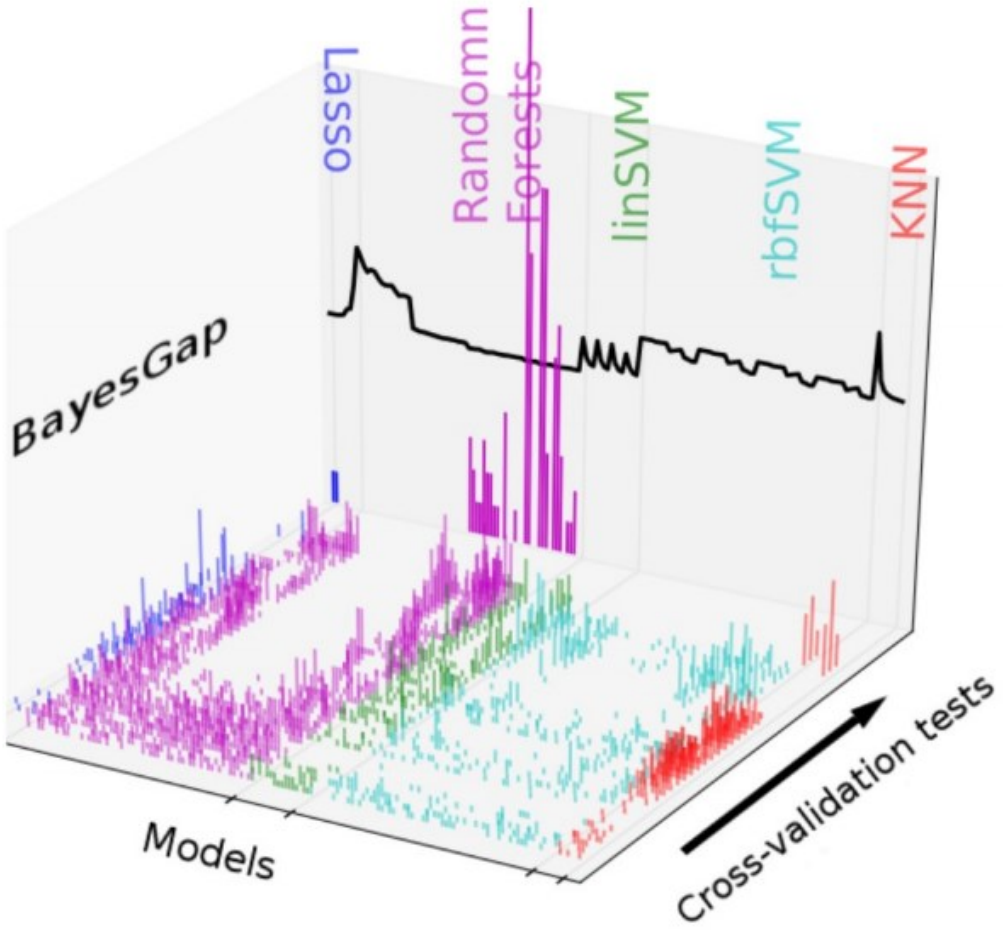


- No closed form expression of the objective
- No access to derivatives
- Possible to obtain observations (possibly noisy) of the function
- Expensive function evaluations
- Non-convex problem

Outline

1. Example
2. Bayesian Optimization Overview
3. More Example Applications
4. Bayesian Optimization Properties, Algorithm, Components
    - Component 1: Gaussian Processes
    - Component 2: Acquisition Functions

Automatic Machine Learning Toolboxes

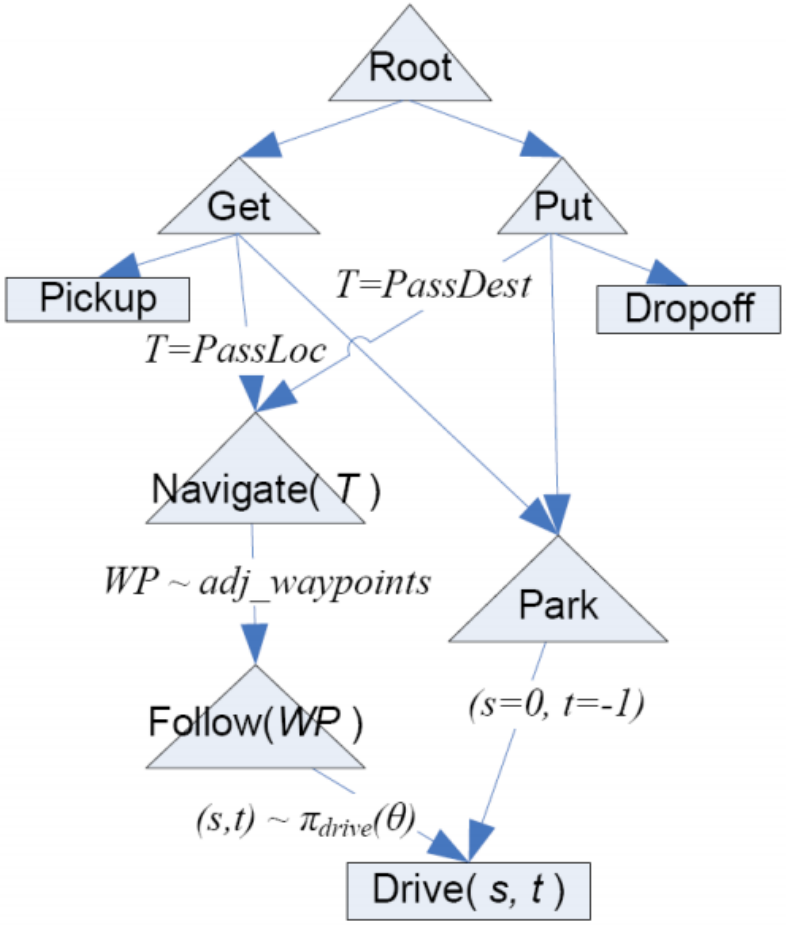


Reinforcement Learning

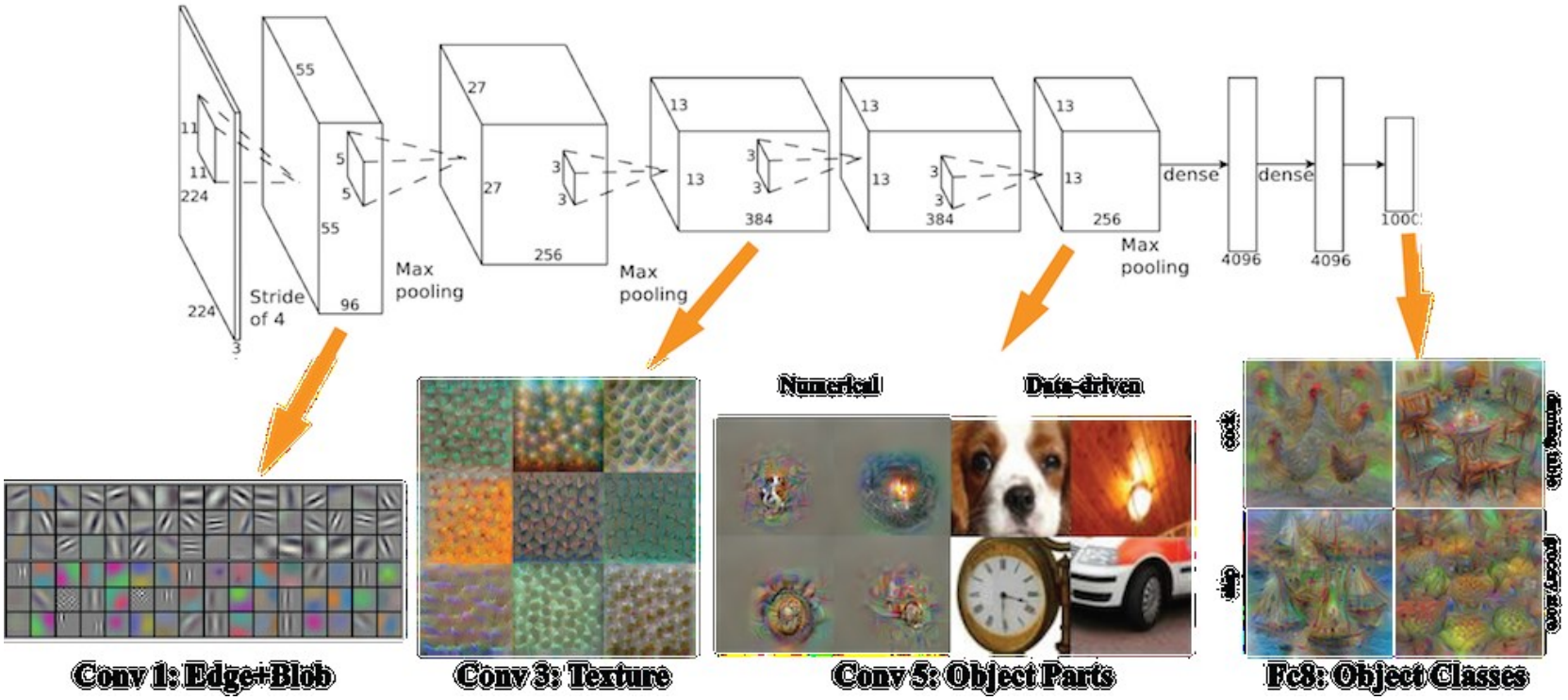
High-level model-based learning for deciding when to navigate, park, pickup and dropoff passengers.

Mid-level active path learning for navigating a topological map.

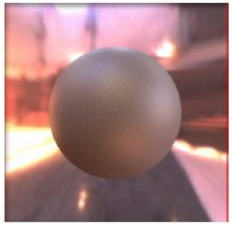
Low-level active policy optimizer to learn control of continuous non-linear vehicle dynamics.



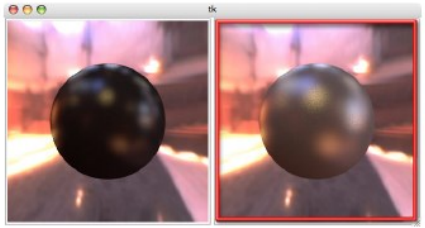
Architecture Configuration in Deep Learning



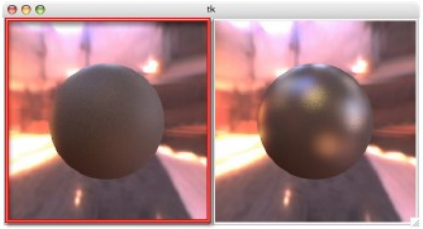
Interactive Animation



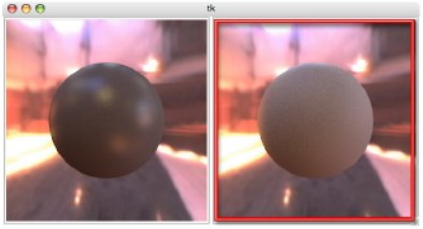
Target



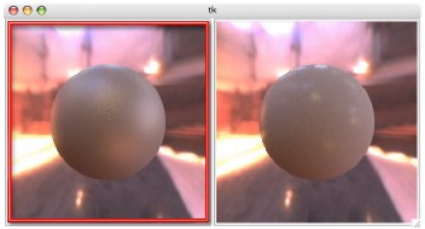
1.



2.



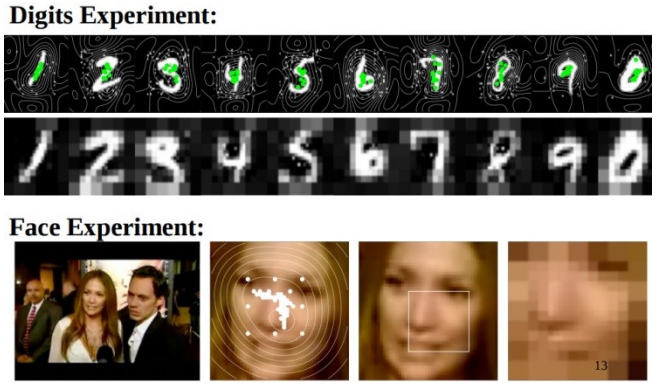
3.

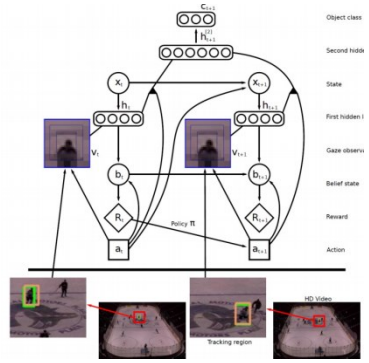


4.

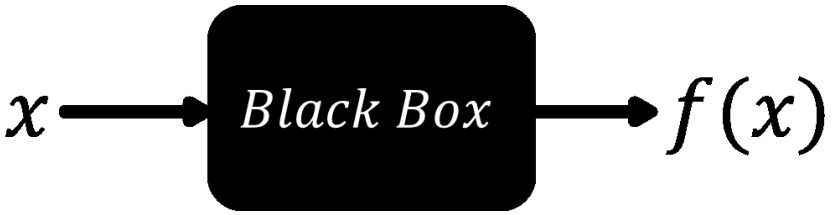
Many Other Applications

- learning to rank
- Robotics
- sensor networks
- automatic algorithm configuration
- GP policy for tracking
- …





Bayesian Optimization



- No closed form expression of the objective
- No access to derivatives
- Possible to obtain observations (possibly noisy) of the function
- Expensive function evaluations
- Non-convex problem

Outline

1. Example
2. Bayesian Optimization Overview
3. More Example Applications
4. Bayesian Optimization Properties, Algorithm, Components
    - Component 1: Gaussian Processes
    - Component 2: Acquisition Functions

How to Bayesian Optimize?

- Uses Bayes Theorem to incorporate prior belief about the problem

- Given
  - Samples $\boldsymbol{x}_i$
  - Noisy Observations $y_i = f(\boldsymbol{x}_i) + \epsilon_i$
  - Prior distribution

Blackbox Objective $f(\boldsymbol{x})$

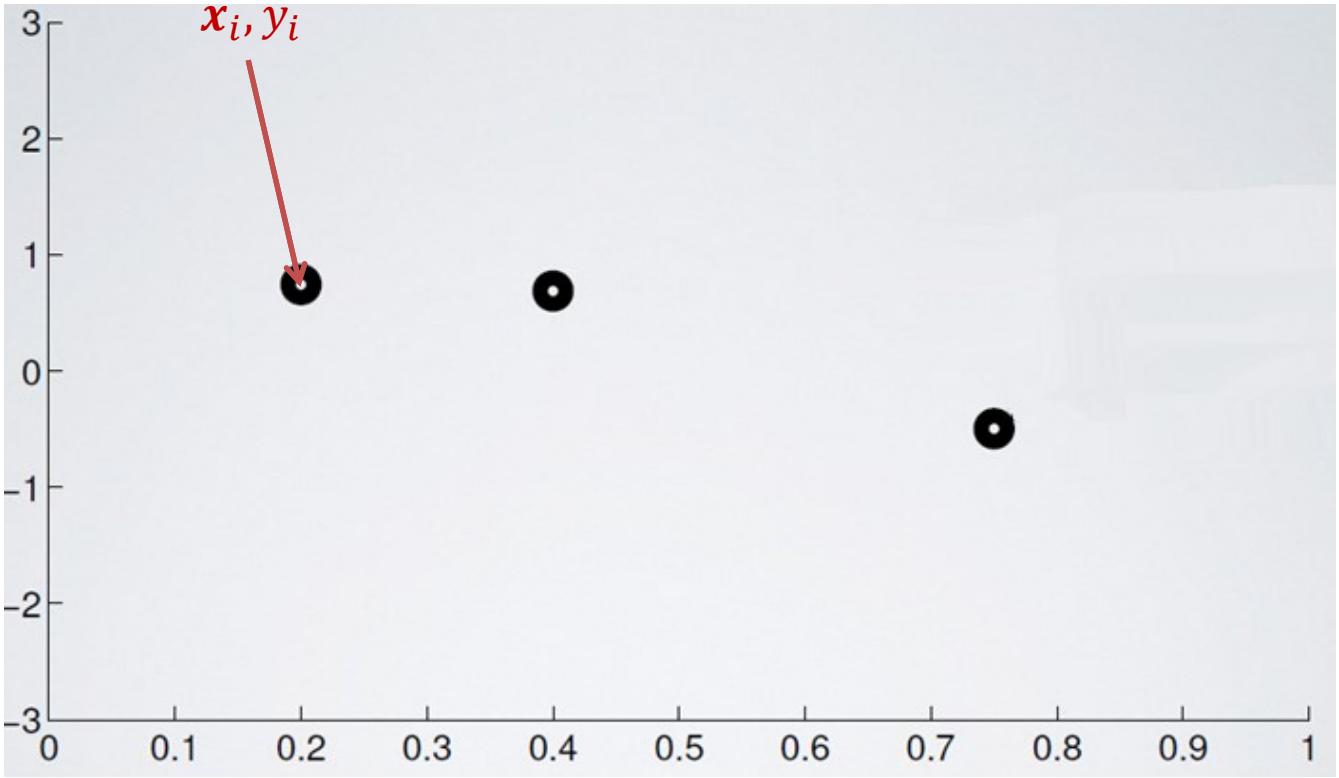
$$P(f|\mathcal{D}_{1:t}) \propto P(\mathcal{D}_{1:t}|f)P(f)$$

$$D_{1:t} = \{\boldsymbol{x}_{1:t}, y_{1:t}\}$$

Posterior Distribution

$$P(f|\mathcal{D}_{1:t}) \propto P(\mathcal{D}_{1:t}|f)P(f)$$

Prior (Gaussian Process)

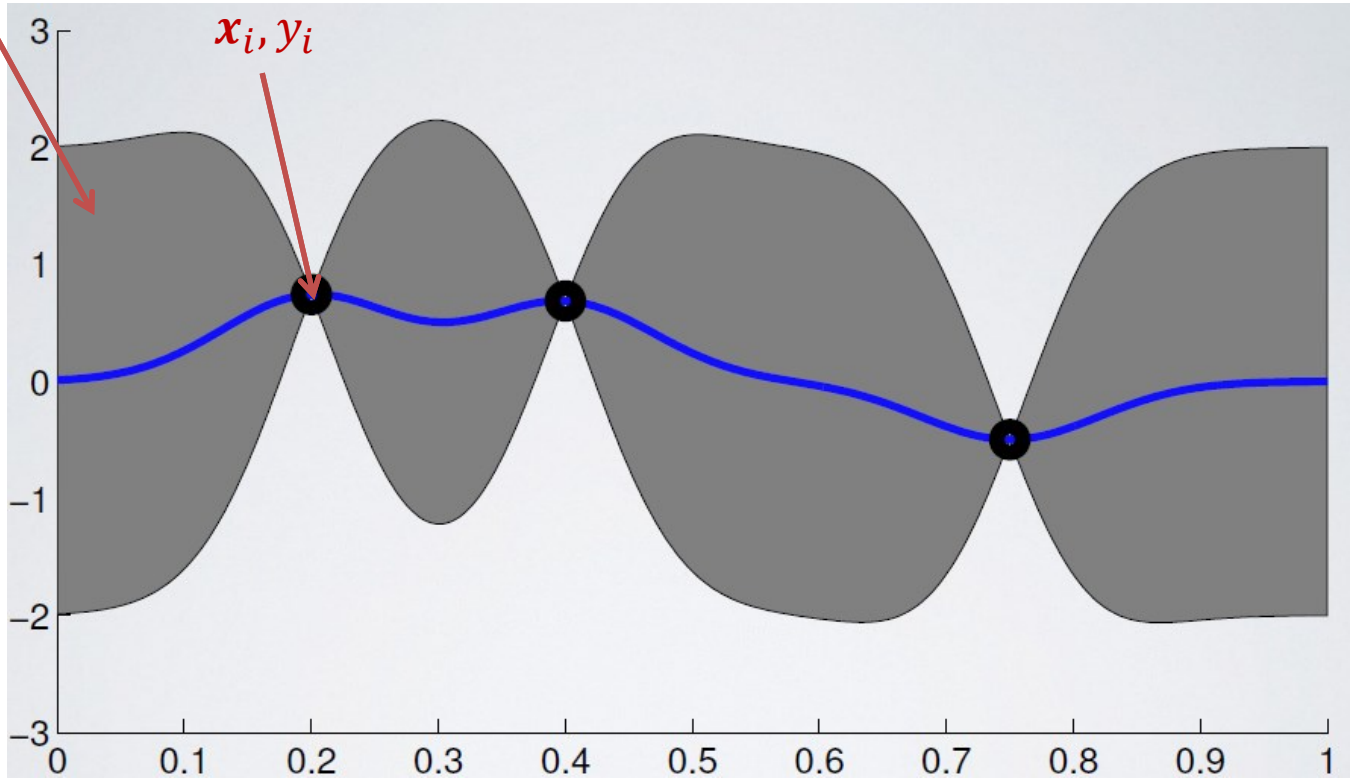


Posterior Distribution

$$P(f|\mathcal{D}_{1:t}) \propto P(\mathcal{D}_{1:t}|f)P(f)$$

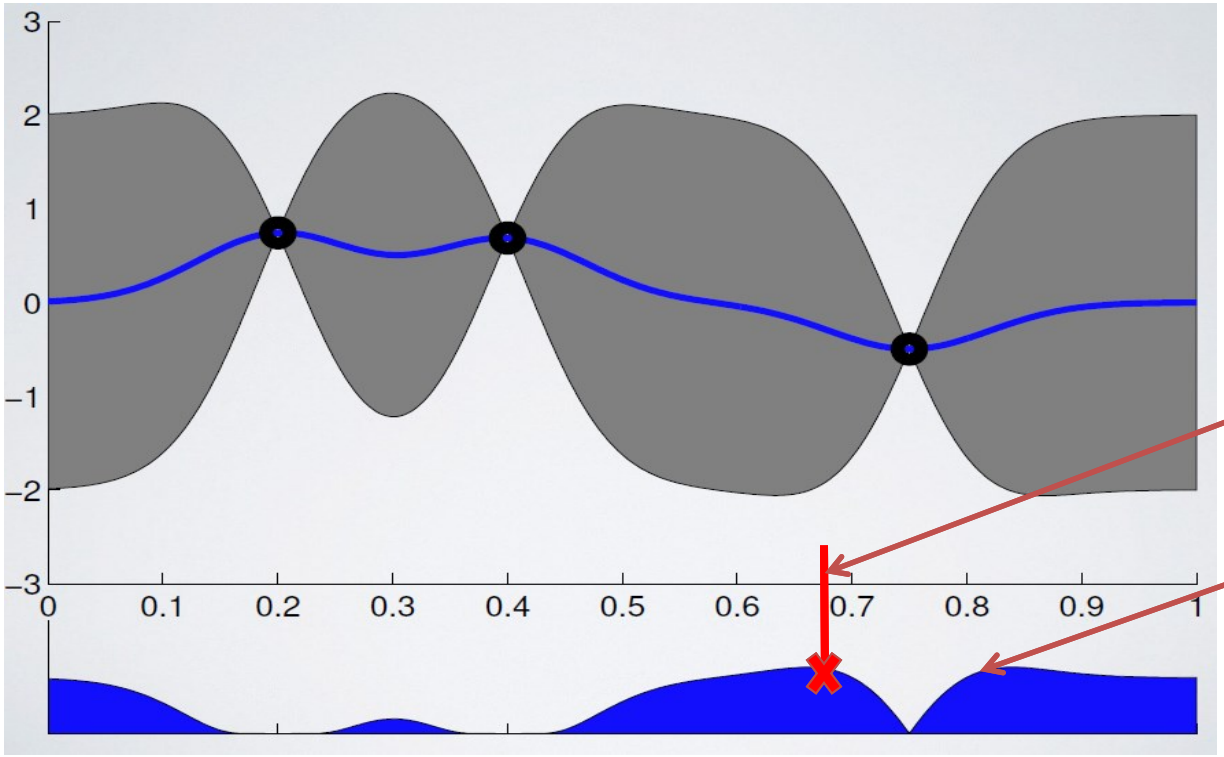
Posterior

Prior (Gaussian Process)

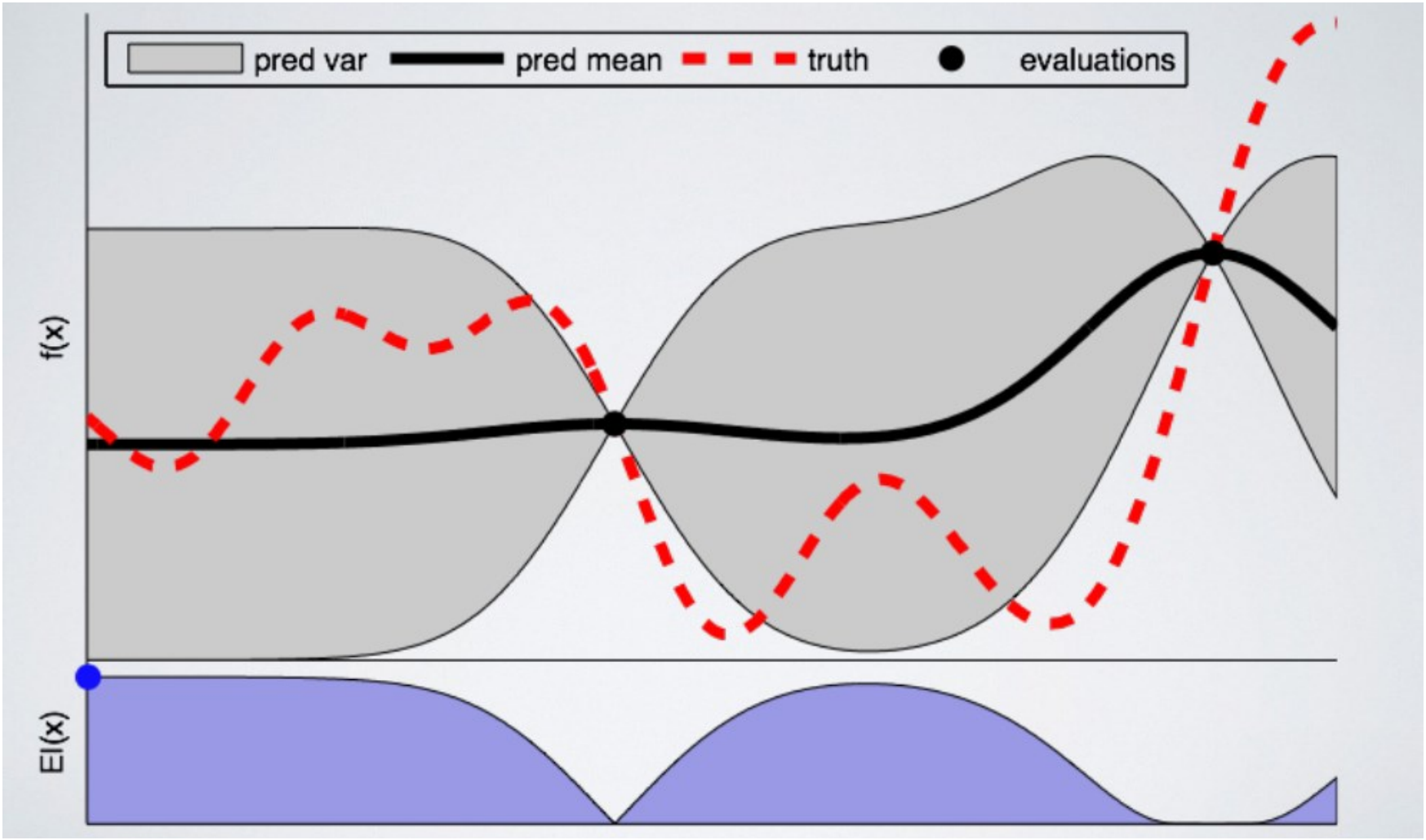


How to Bayesian Optimize?

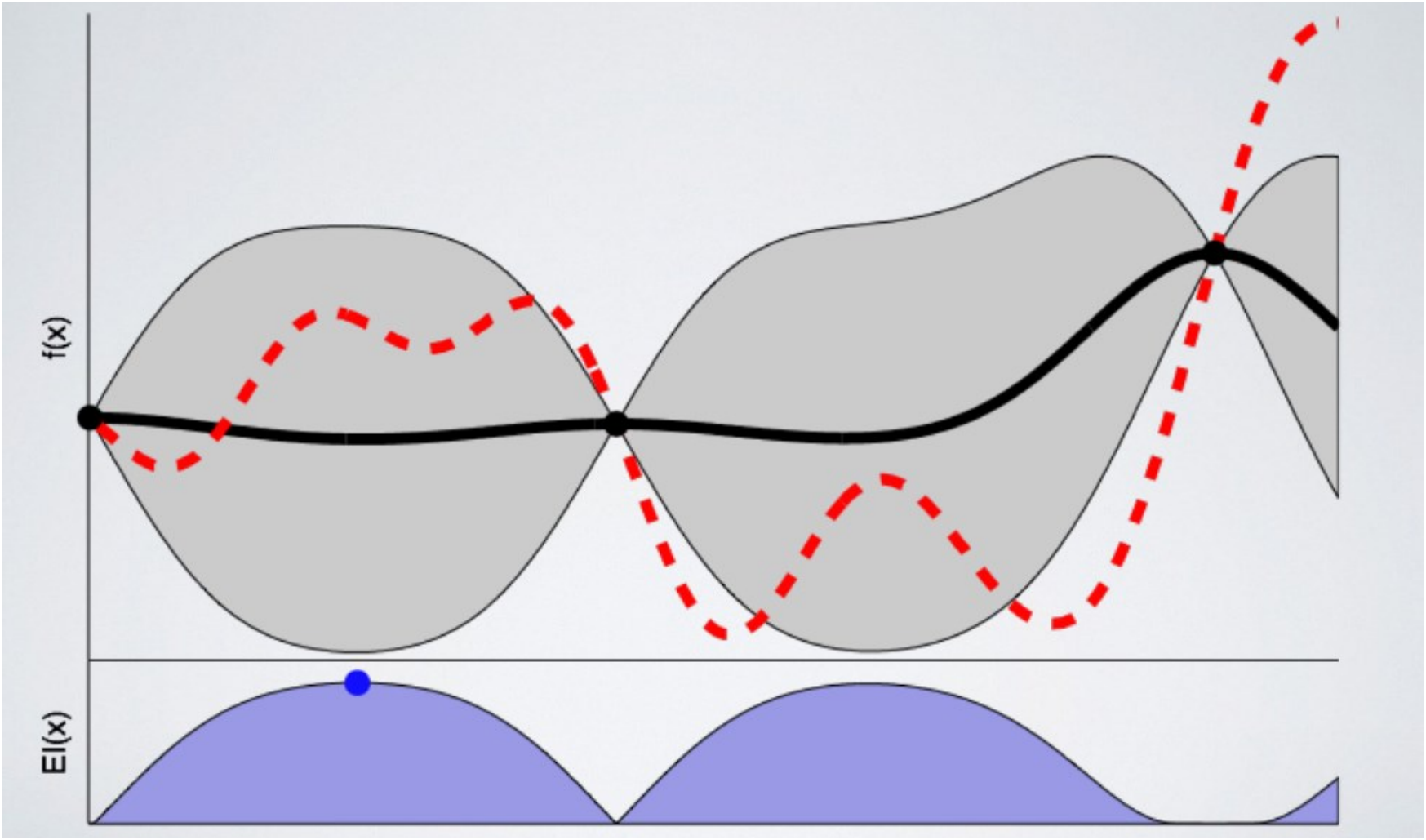
- Uses an **acquisition function** to determine next location



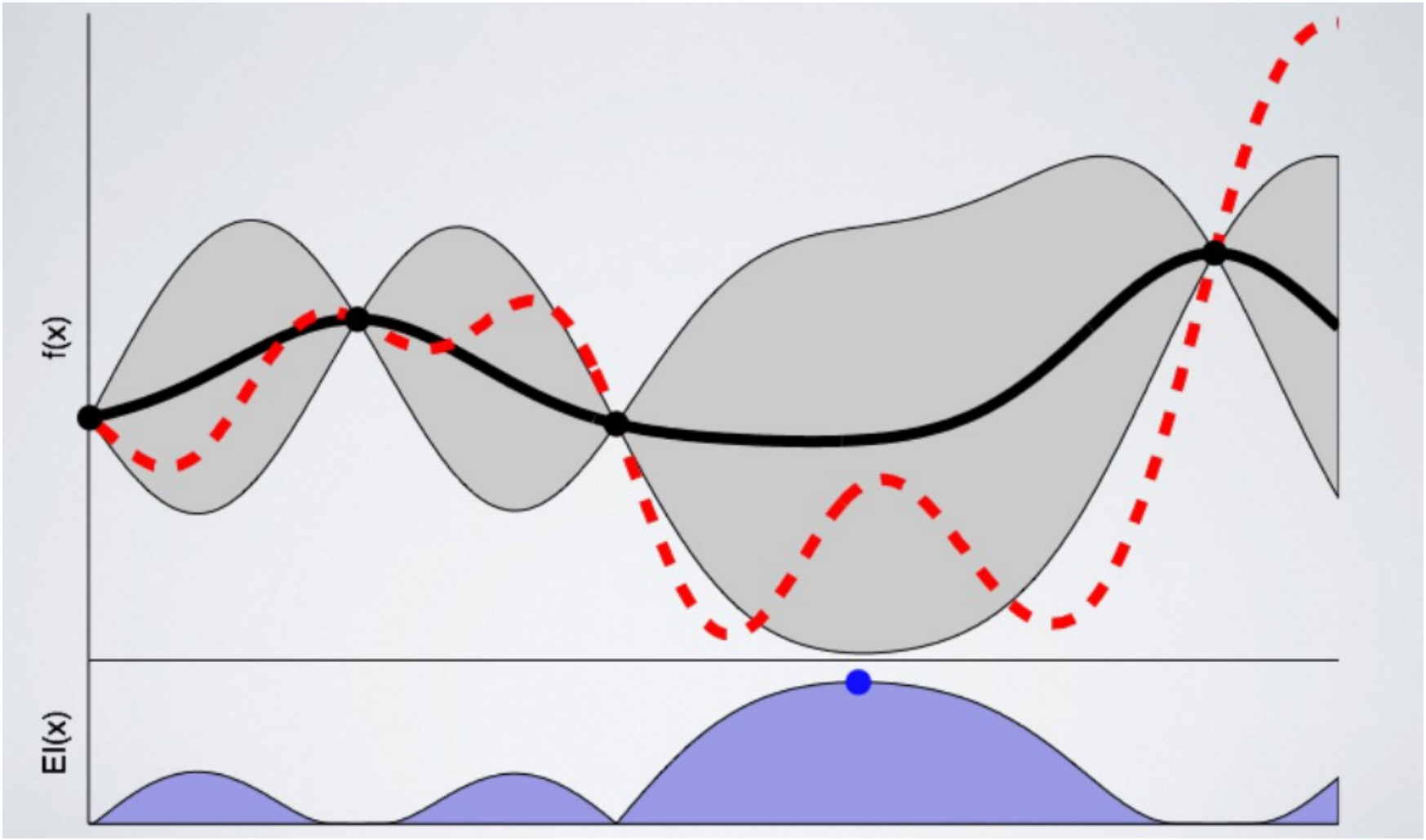
How to Bayesian Optimize?



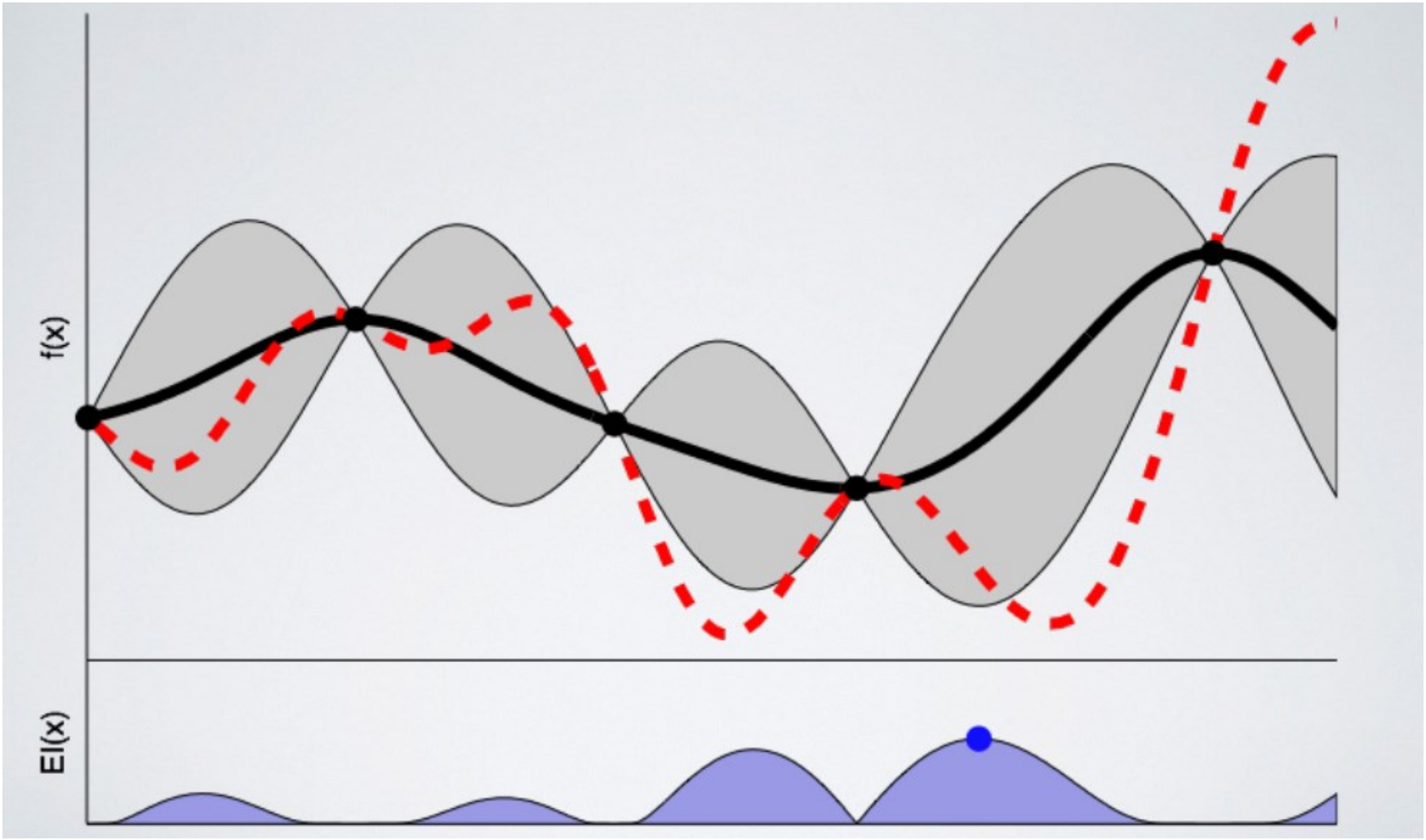
How to Bayesian Optimize?



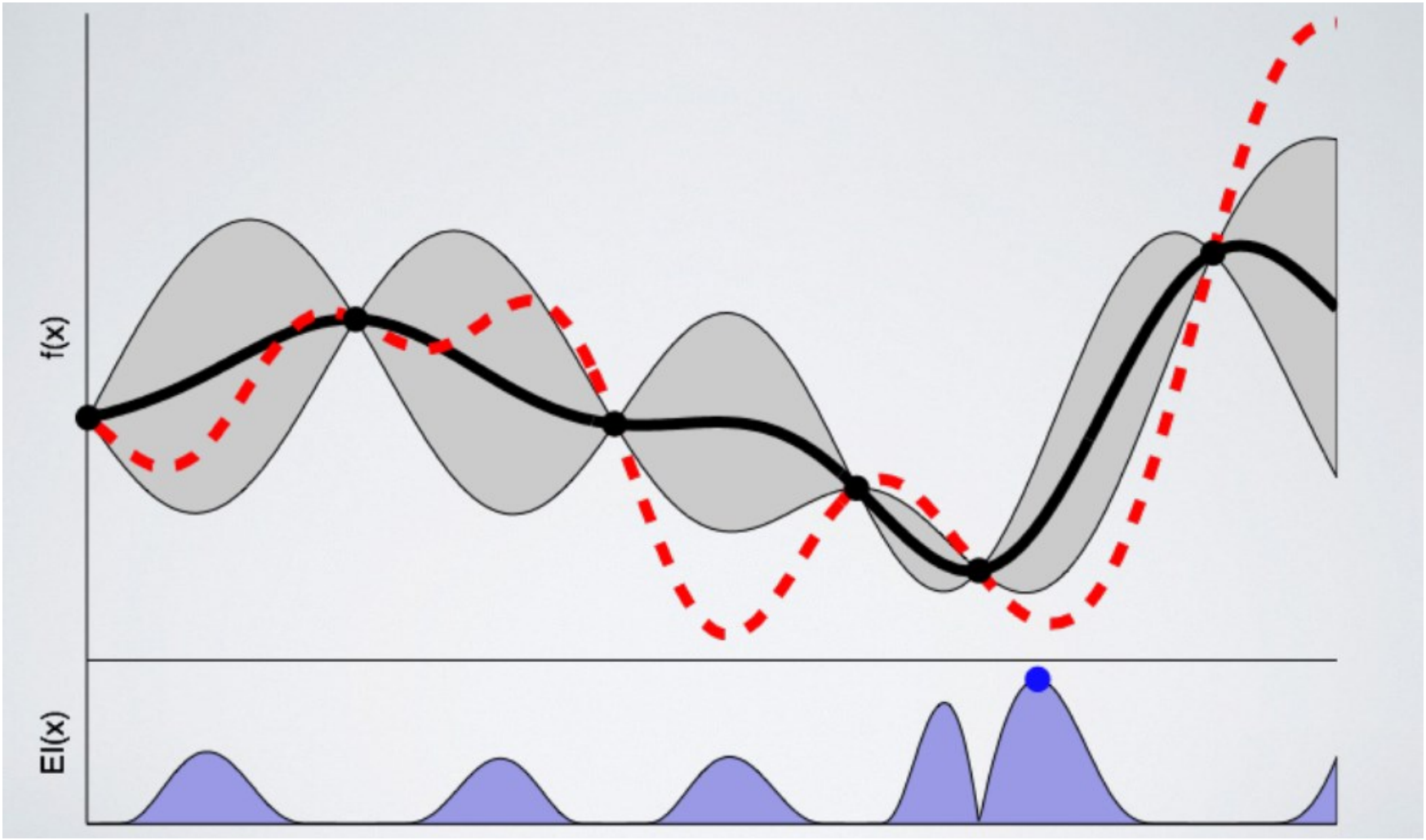
How to Bayesian Optimize?



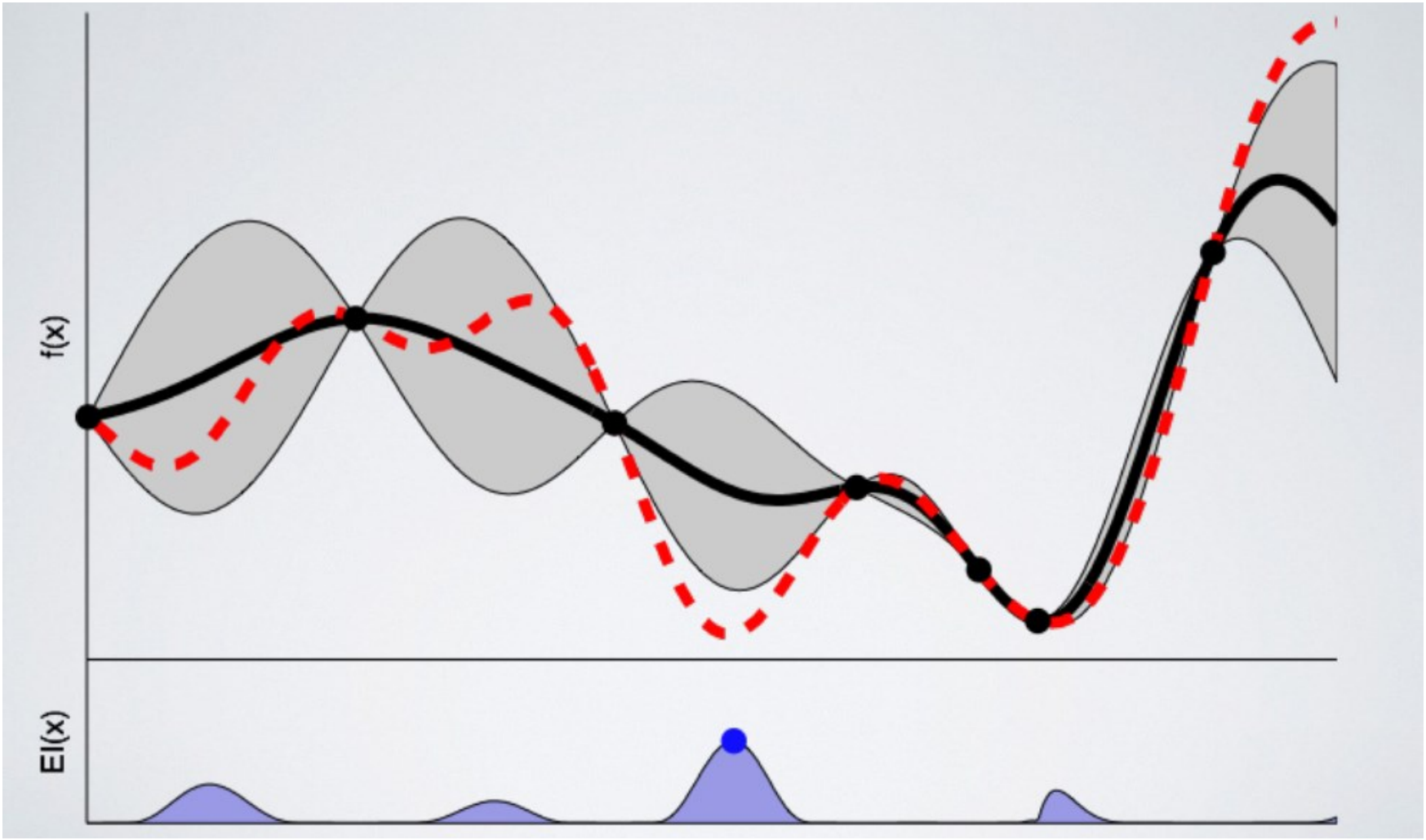
How to Bayesian Optimize?



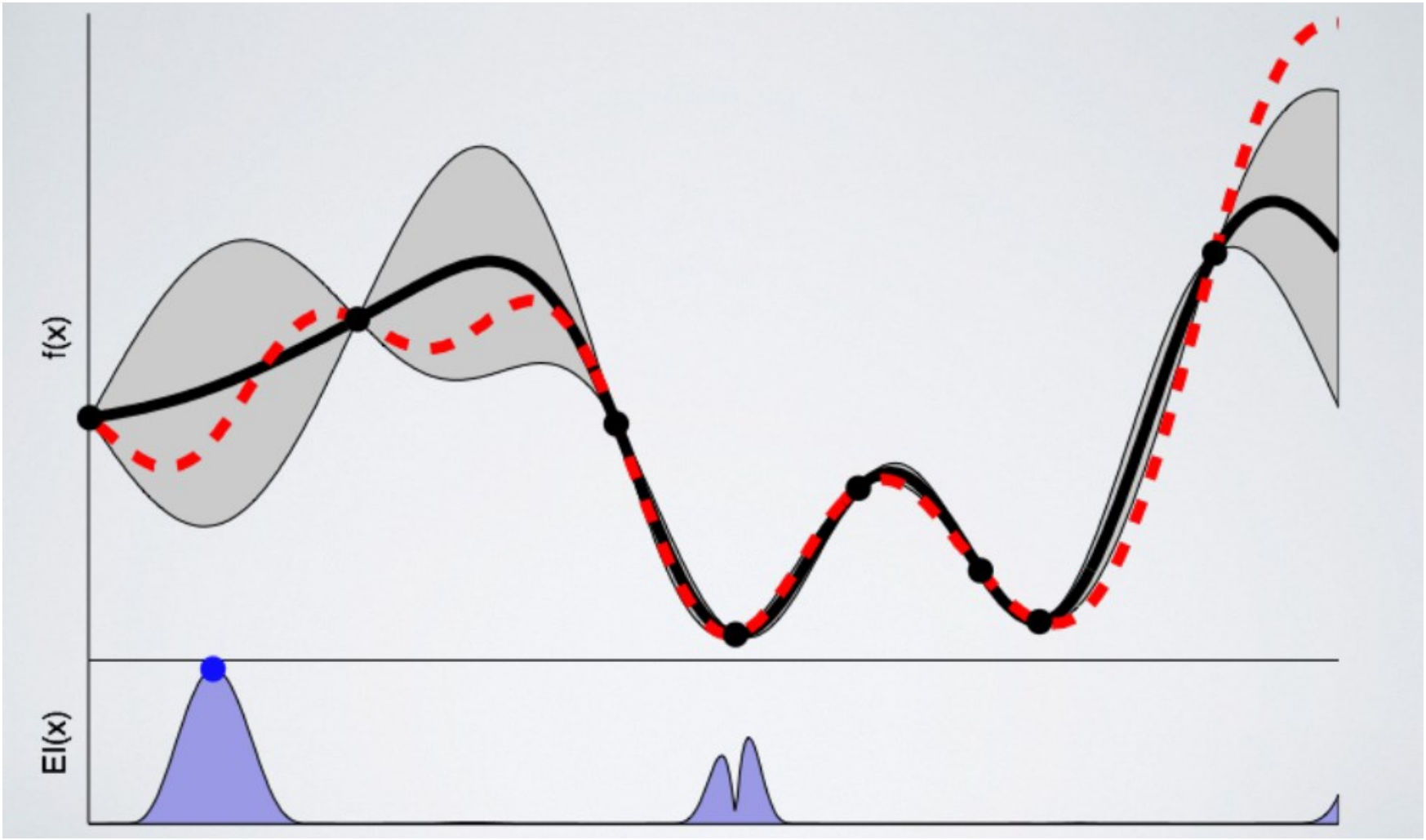
How to Bayesian Optimize?



How to Bayesian Optimize?



How to Bayesian Optimize?



How to Bayesian Optimize?

Algorithm 1 Bayesian Optimization

1: **for** $t = 1, 2, \ldots$ **do**
2: Find $\mathbf{x}_t$ by optimizing the acquisition function over the GP: $\mathbf{x}_t = \text{argmax}_{\mathbf{x}} u(\mathbf{x}|\mathcal{D}_{1:t-1})$.
3: Sample the objective function: $y_t = f(\mathbf{x}_t) + \varepsilon_t$.
4: Augment the data $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (\mathbf{x}_t, y_t)\}$ and update the GP.
5: **end for**

- Sequential decision making process
  - How to estimate and update the GP?
  - How to define the acquisition function?

Outline

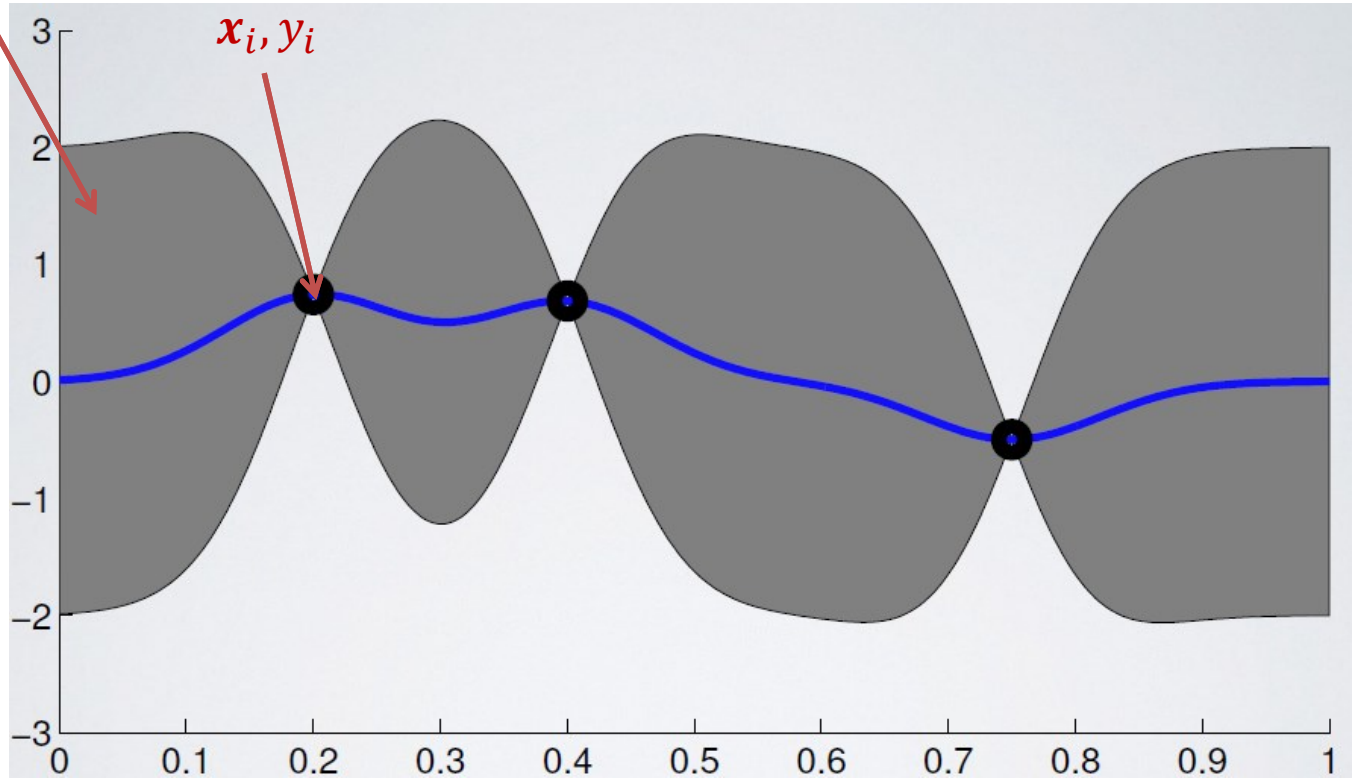
1. Example
2. Bayesian Optimization Overview
3. More Example Applications
4. Bayesian Optimization Properties, Algorithm, Components
    - Component 1: Gaussian Processes
    - Component 2: Acquisition Functions

Gaussian Process Prior

$$P(f|\mathcal{D}_{1:t}) \propto P(\mathcal{D}_{1:t}|f)P(f)$$

Posterior

Prior (Gaussian Process)



GP Definition

Gaussian Distribution is a distribution over a random variable, completely specified by its mean $\boldsymbol{\mu}$, and covariance Σ

$$\mathbf{x} \sim N(\boldsymbol{\mu}, \Sigma)$$

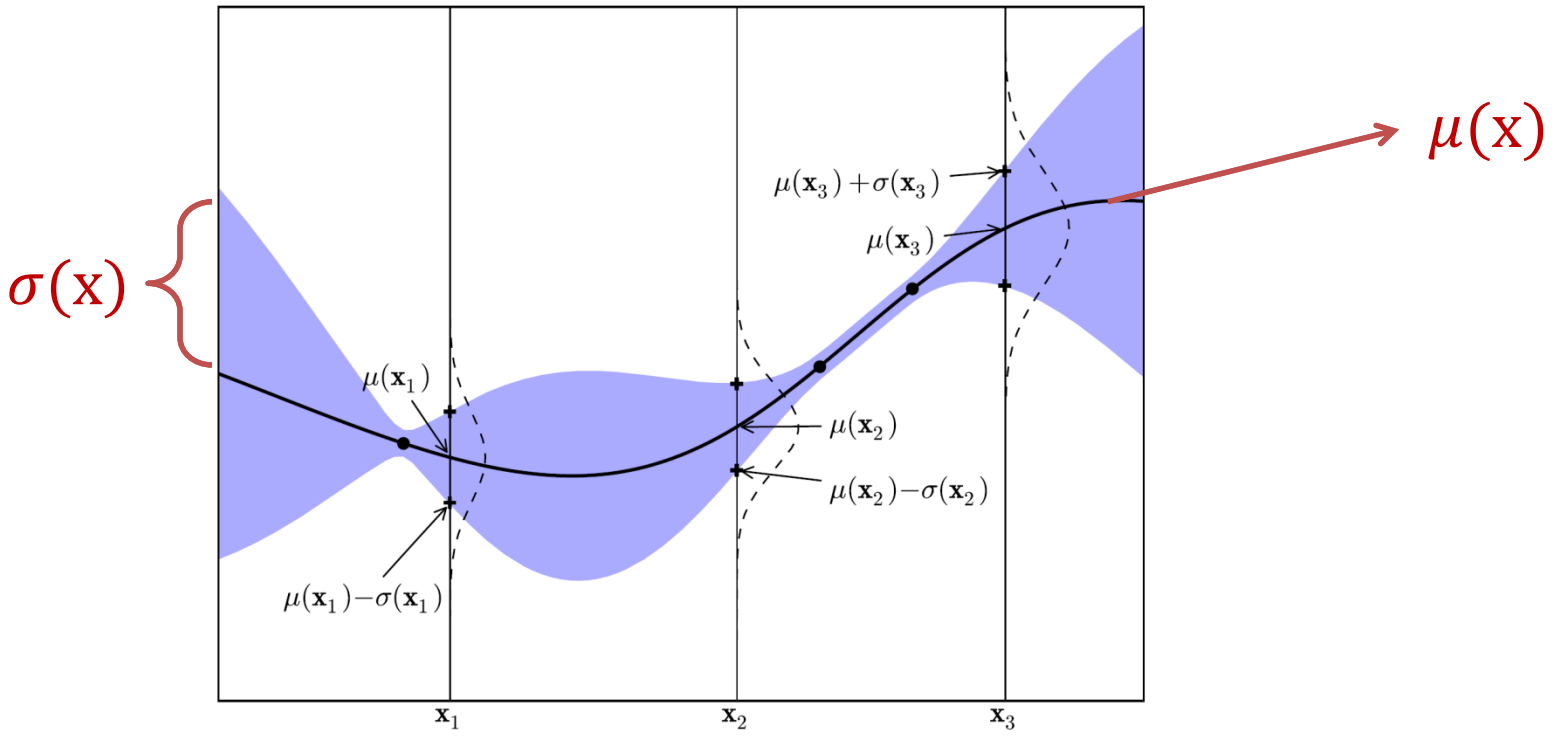
Gaussian Process is a distribution over a function, completely specified by its mean function m, and covariance function k

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

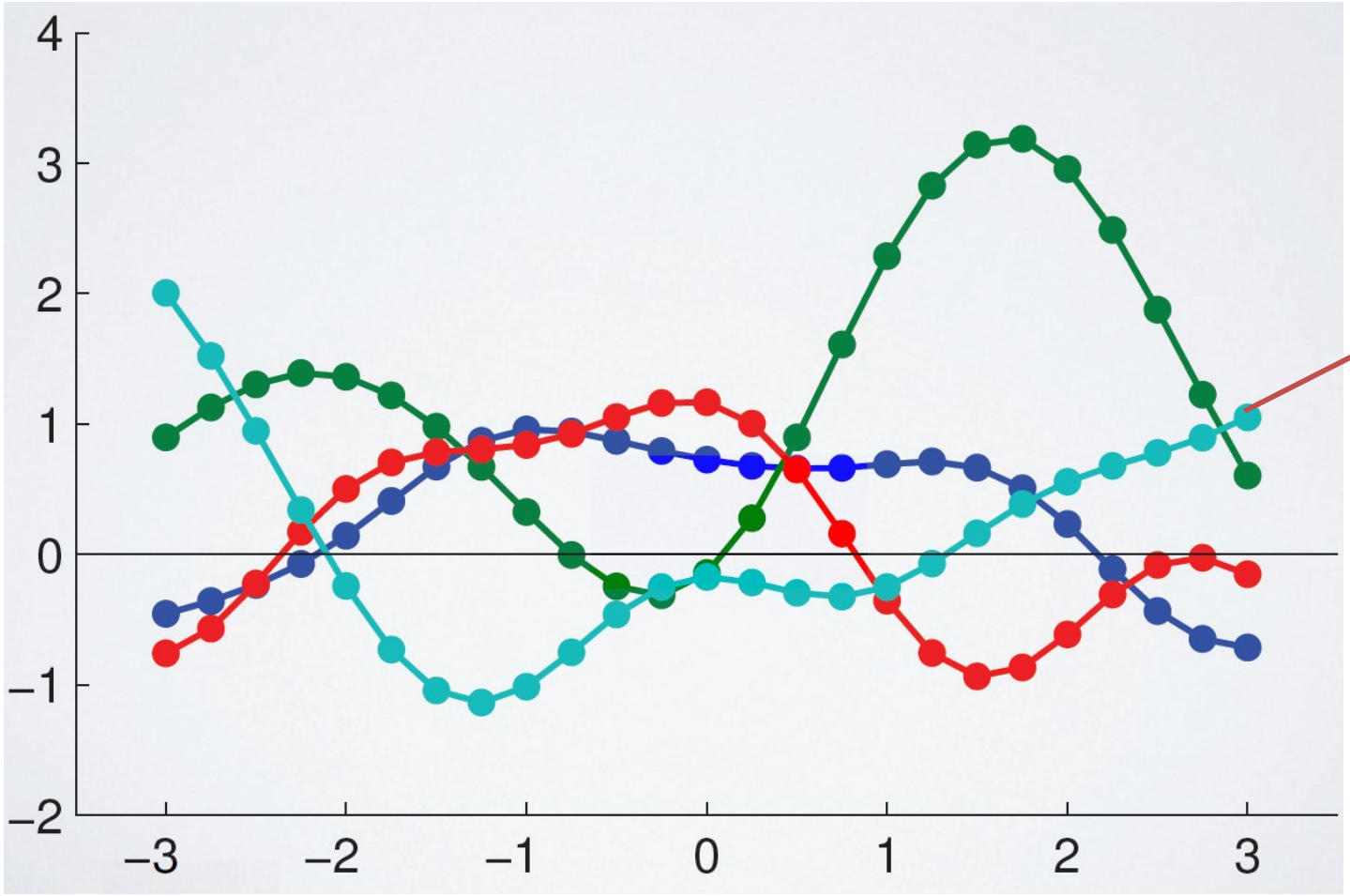
GP Intuitively

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Given $\mathbf{x}$, instead of returning a scalar $f(\mathbf{x})$, it returns the mean and variance of a normal distribution over possible values of f at $\mathbf{x}$.



GP Intuitively



GP Components

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Assuming prior mean is zero function, any finite combination of dimensions will be a Gaussian distribution

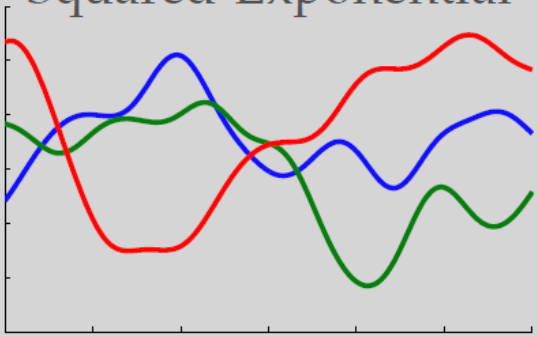
$$f(\mathbf{x}_{1:t}) \sim N(\mathbf{0}, \mathbf{K})$$

Kernel matrix

$$m(\mathbf{x}) = \mathbf{0} \qquad \mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_t, \mathbf{x}_1) & \dots & k(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix}$$

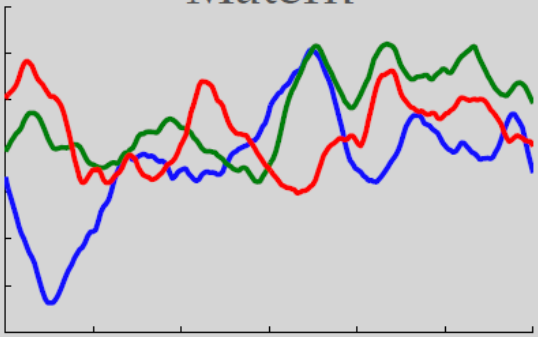
Covariance Function Examples

Squared-Exponential



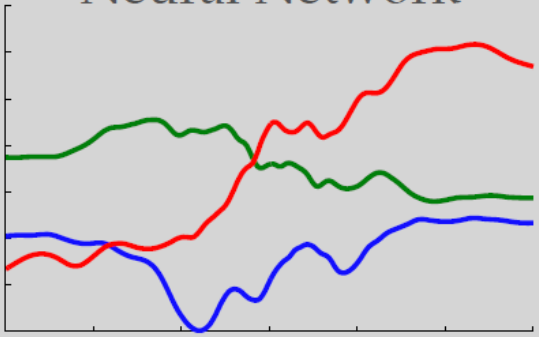
$$C(x, x') = \alpha \exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_d - x'_d}{\ell_d}\right)^2\right\}$$

Matérn



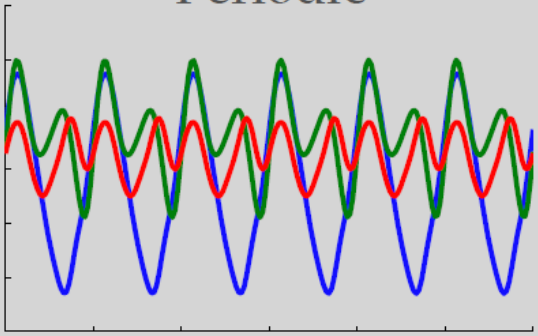
$$C(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\, r}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}\, r}{\ell}\right)$$

"Neural Network"



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

Periodic



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

GP Fitting

- In the Bayesian Optimization task, we use data from external model to fit a GP and get the posterior.

- Given
  - Samples $\boldsymbol{x}_i$
  - Observations $y_i = f(\boldsymbol{x}_i) + \epsilon_i$
  - Prior distribution

$$P(f|\mathcal{D}_{1:t}) \propto P(\mathcal{D}_{1:t}|f)P(f)$$

$\mu(x), \sigma^2(\mathrm{x})$ $\quad$ $D_{1:t} = \{\boldsymbol{x}_{1:t}, y_{1:t}\}$

GP Fitting

$$\mathbf{f}_{1:t} \sim N(\mathbf{0}, \mathbf{K}) \longrightarrow \text{Prior}$$

For a point $\mathbf{x}^*$,
$f^* = f(\mathbf{x}^*)$ and $\mathbf{f}_{1:t}$ are jointly Gaussian (Property of GP)

$$\begin{bmatrix} \mathbf{f}_{1:t} \\ f^* \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right)$$

where $\mathbf{k} = [k(\mathbf{x}^*, \mathbf{x}_1) \quad k(\mathbf{x}^*, \mathbf{x}_2) \quad \dots \quad k(\mathbf{x}^*, \mathbf{x}_1)]^T$

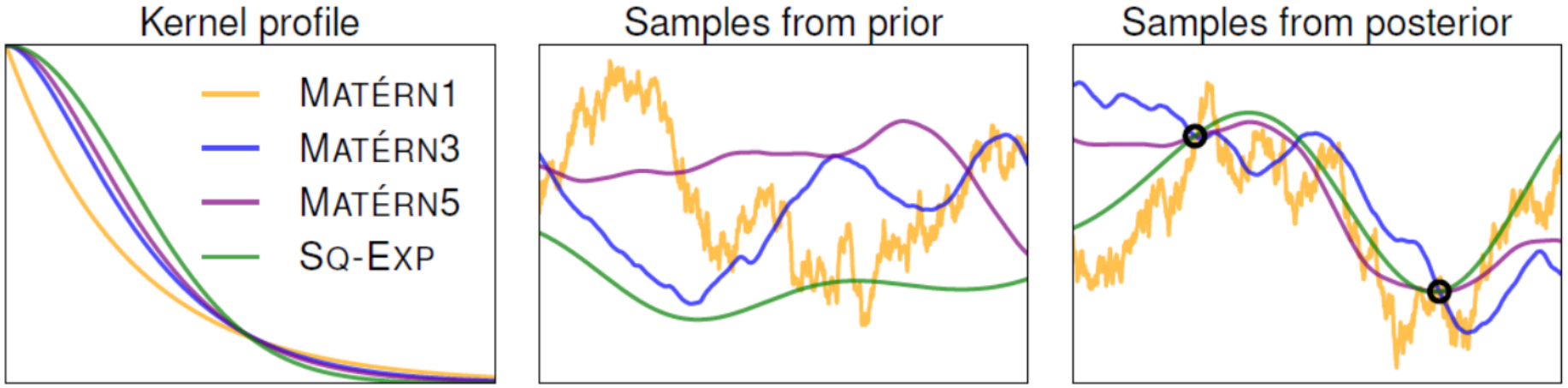
Gaussian Processes (GP)

$$P(f^*|D_{1:t}, x^*) = N\left(\mu(x^*), \sigma^2(x^*)\right)$$

Posterior

$$\mu(x^*) = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{f}_{1:t}$$

$$\sigma^2(x^*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}$$



Noisy Observations

$$y_i = f(\mathbf{x}_i) + \epsilon_i \qquad \epsilon \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$$

Prior

$$f(\mathbf{x}_{1:t}) \sim N(\mathbf{0}, \mathbf{K})$$

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_t, \mathbf{x}_1) & \dots & k(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix} + \sigma^2_{\text{noise}} I$$

Posterior

$$\mu_t(\mathbf{x}_{t+1}) = \mathbf{k}^T [\mathbf{K} + \sigma^2_{\text{noise}} I]^{-1} \mathbf{y}_{1:t}$$

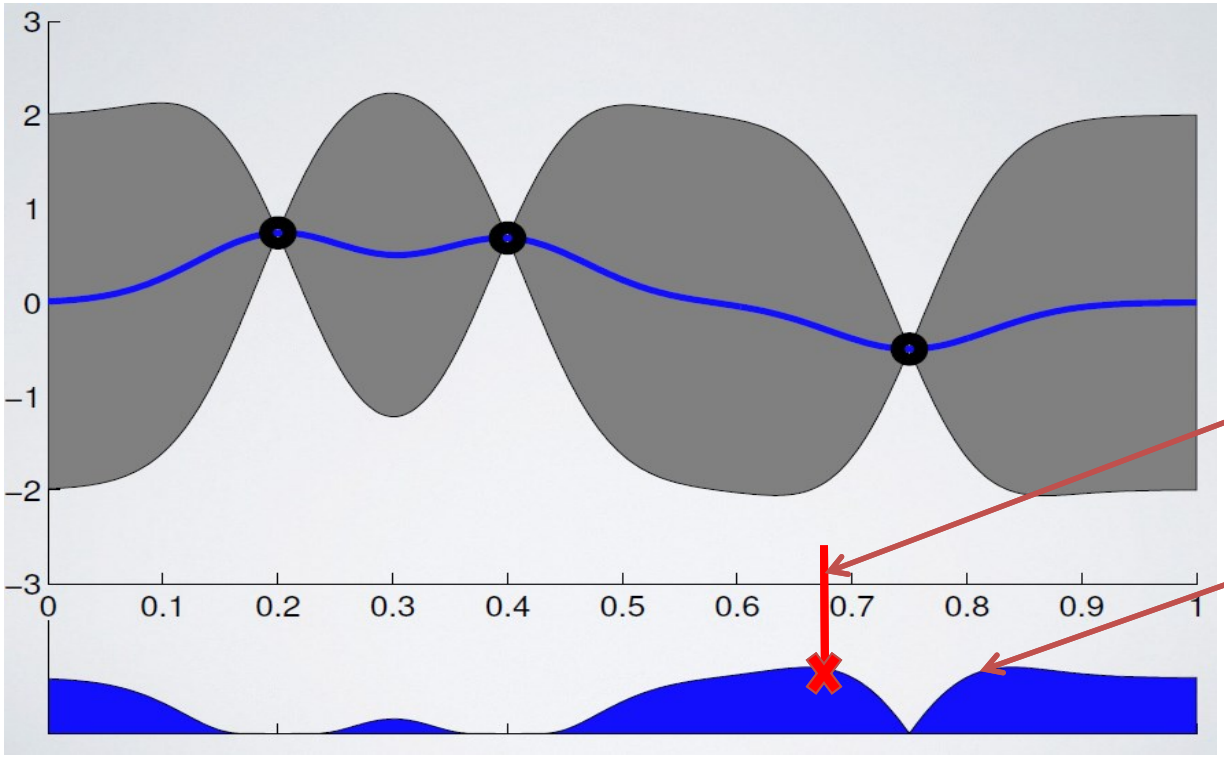
$$\sigma^2_t(\mathbf{x}_{t+1}) = k(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}^T [\mathbf{K} + \sigma^2_{\text{noise}} I]^{-1} \mathbf{k}$$

Outline

1. Example
2. Bayesian Optimization Overview
3. More Example Applications
4. Bayesian Optimization Properties, Algorithm, Components
    - Component 1: Gaussian Processes
    - Component 2: Acquisition Functions

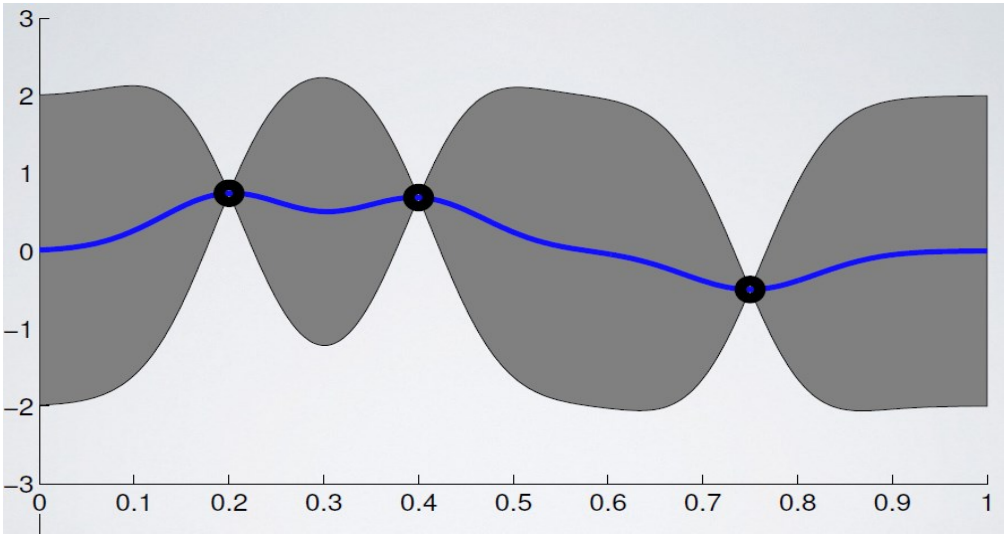
How to Bayesian Optimize?

- Uses an **acquisition function** to determine next location



Exploitation vs. Exploration

- Exploitation
  - Trying values where the objective function is expected to be high
  - Sample points close to maximum $\mu(x)$
- Exploration
  - Trying values where the objective function is very uncertain
  - Sample points where $\sigma(x)$ is maximum

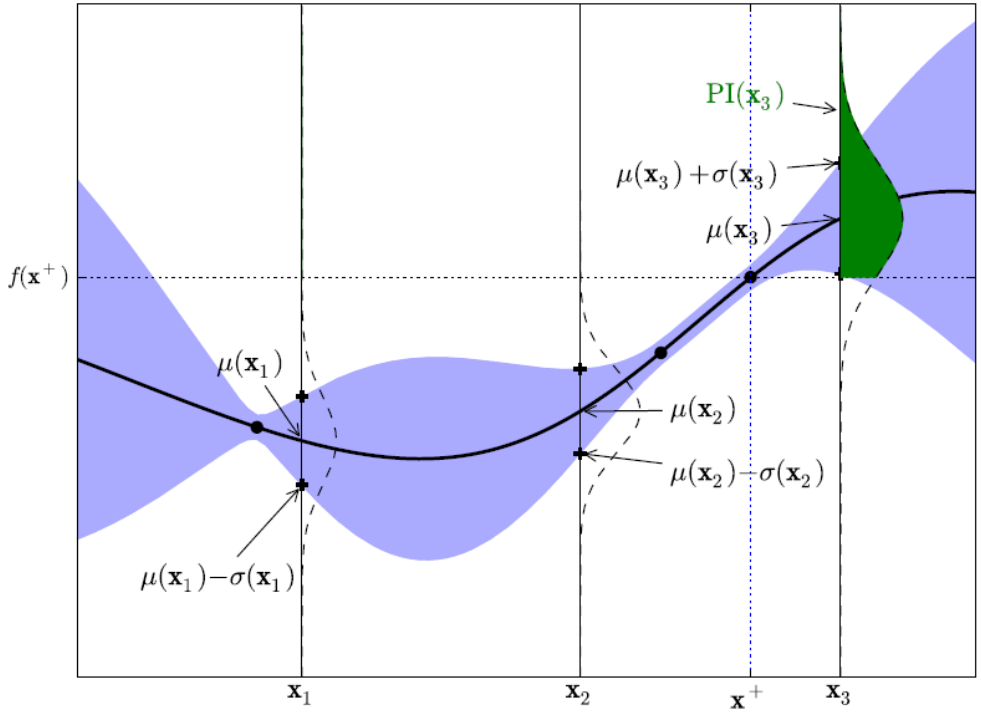


Acquisition Functions

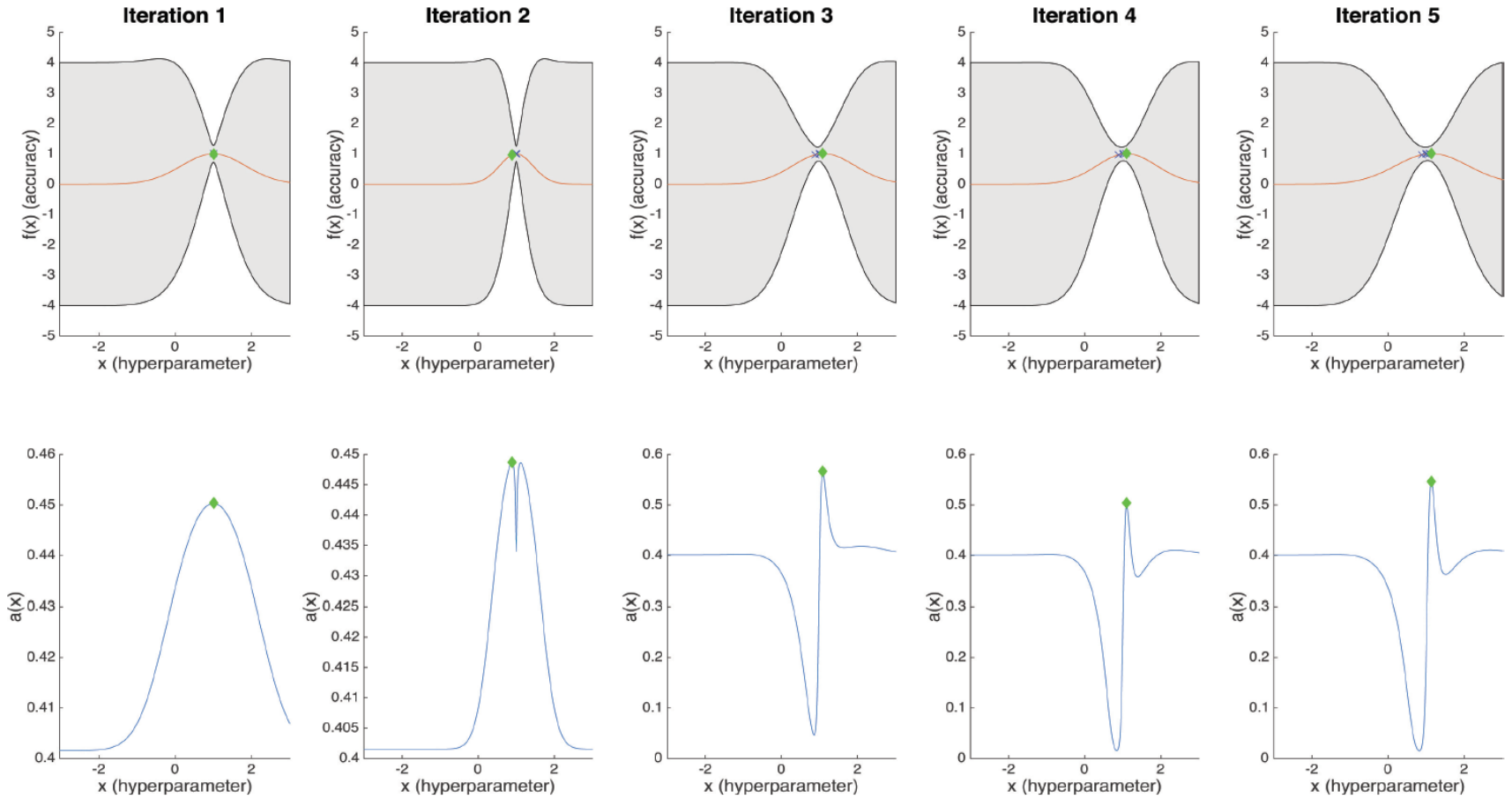
- Probability of Improvement (PI)
- Expected Improvement (EI)
- Upper Confidence Bounds (UCB)
- Thompson Sampling
- Mixtures of above functions

Probability of Improvement (PI)

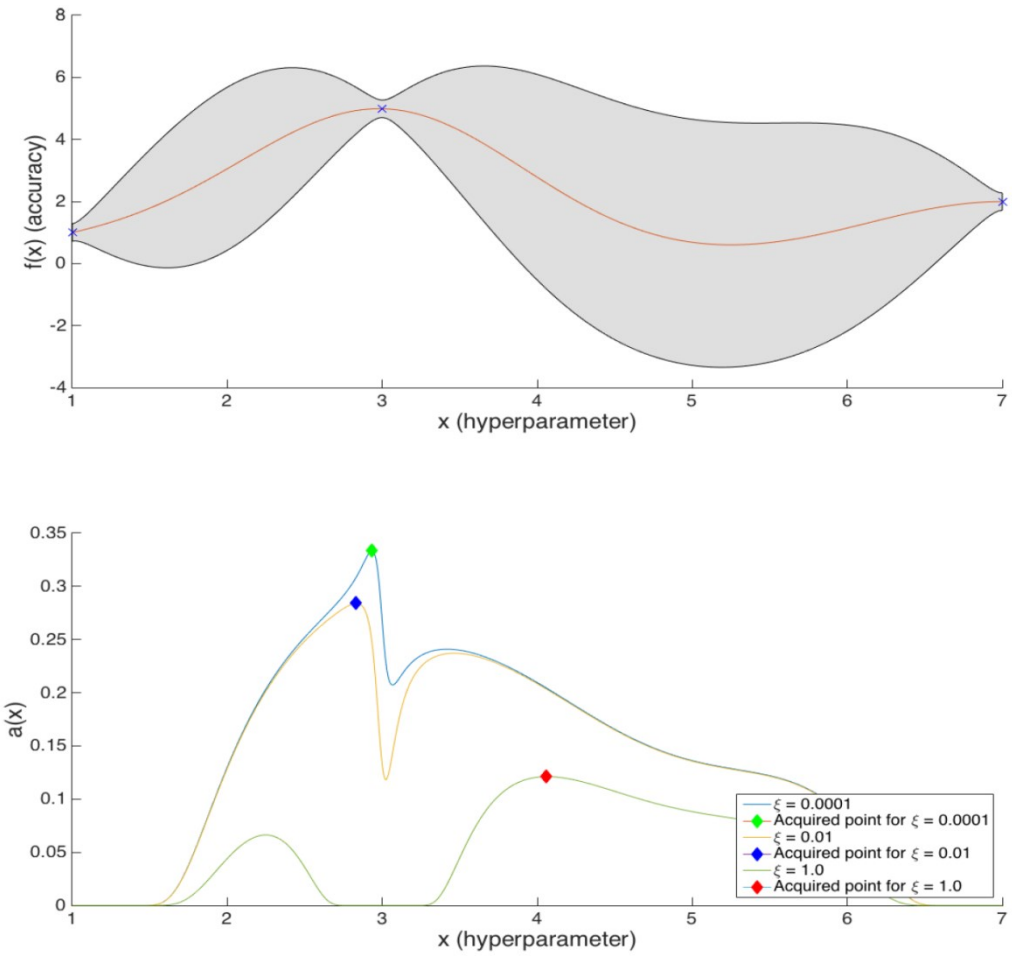
$$\begin{aligned} \mathrm{PI}(\mathbf{x}) &= P(f(\mathbf{x}) \geq f(\mathbf{x}^+) + \xi) \\ &= \Phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})}\right) \end{aligned}$$



Example of PI



PI Exploration-Exploitation Trade-off



Balancing exploration and exploitation is biased to exploitation.

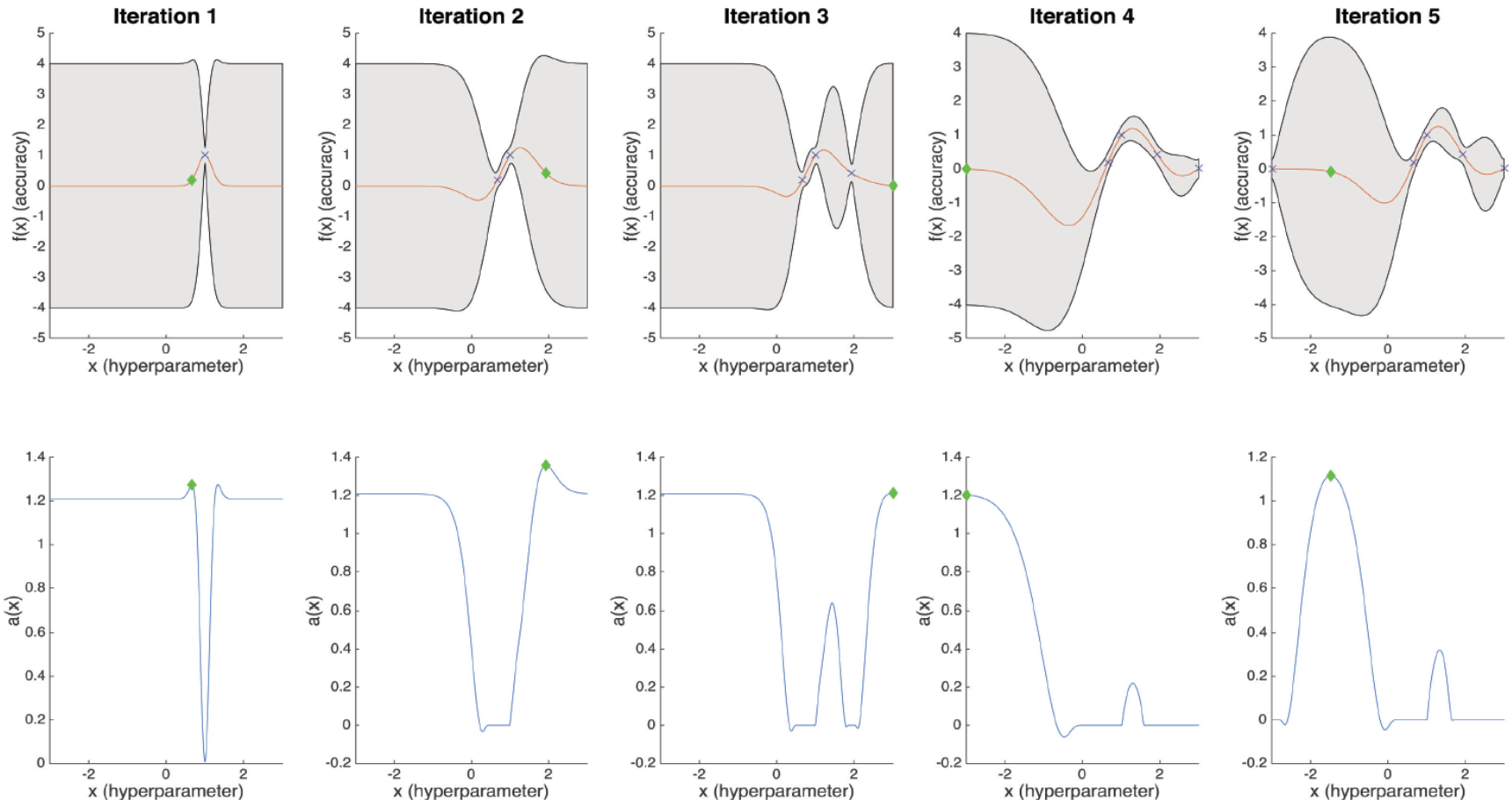
Expected Improvement (EI)

$$\mathrm{EI}(\mathbf{x}) = \mathbb{E}(\max\{0, f_{t+1}(\mathbf{x}) - f(\mathbf{x}^+)\} \mid \mathcal{D}_t)$$

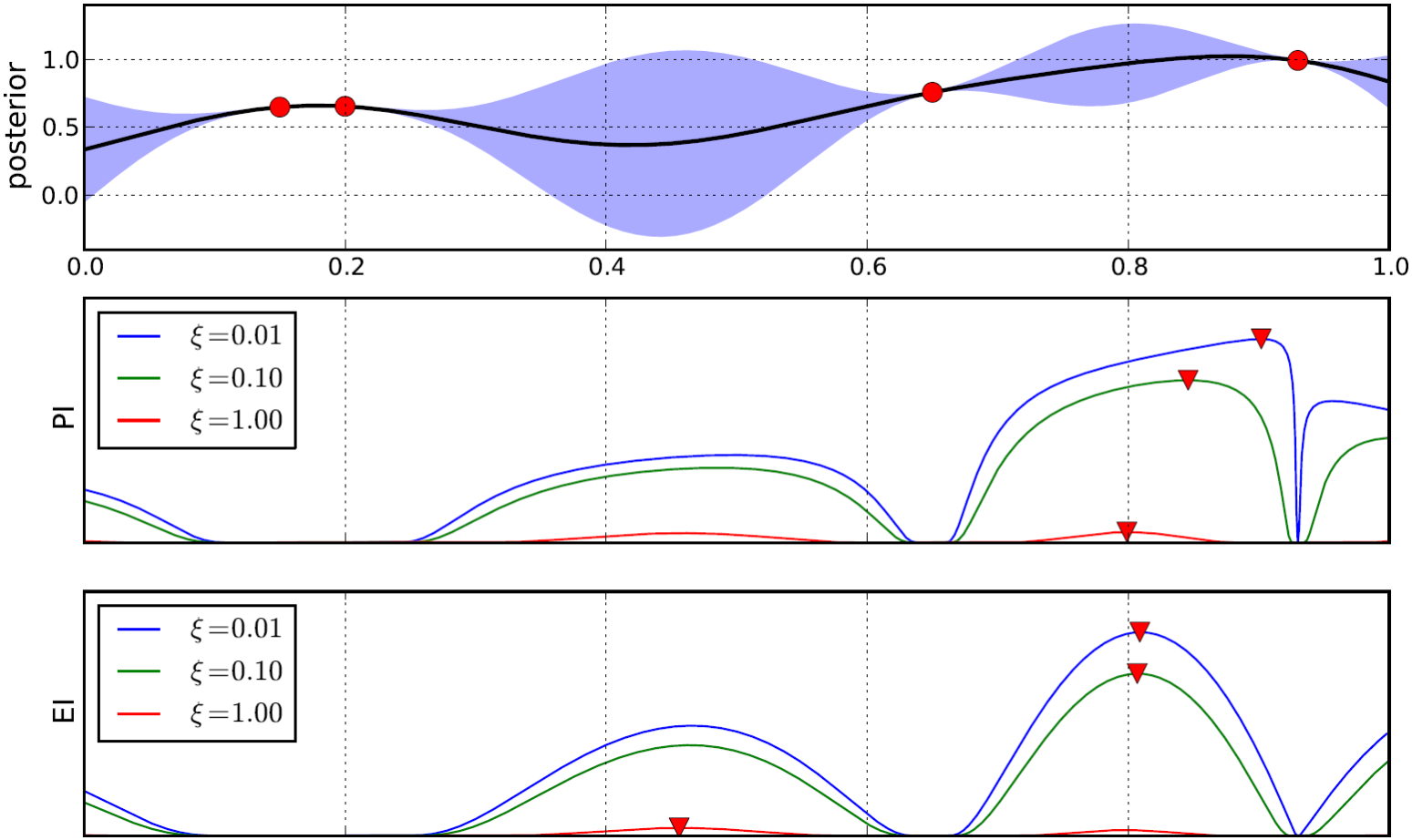
$$\mathrm{EI}(\mathbf{x}) = \begin{cases} (\mu(\mathbf{x}) - f(\mathbf{x}^+))\Phi(Z) + \sigma(\mathbf{x})\phi(Z) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases}$$

$$Z = \begin{cases} \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})} & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases}$$

Example of EI



PI vs. EI


posterior
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0
PI
$\xi=0.01$
$\xi=0.10$
$\xi=1.00$
EI
$\xi=0.01$
$\xi=0.10$
$\xi=1.00$

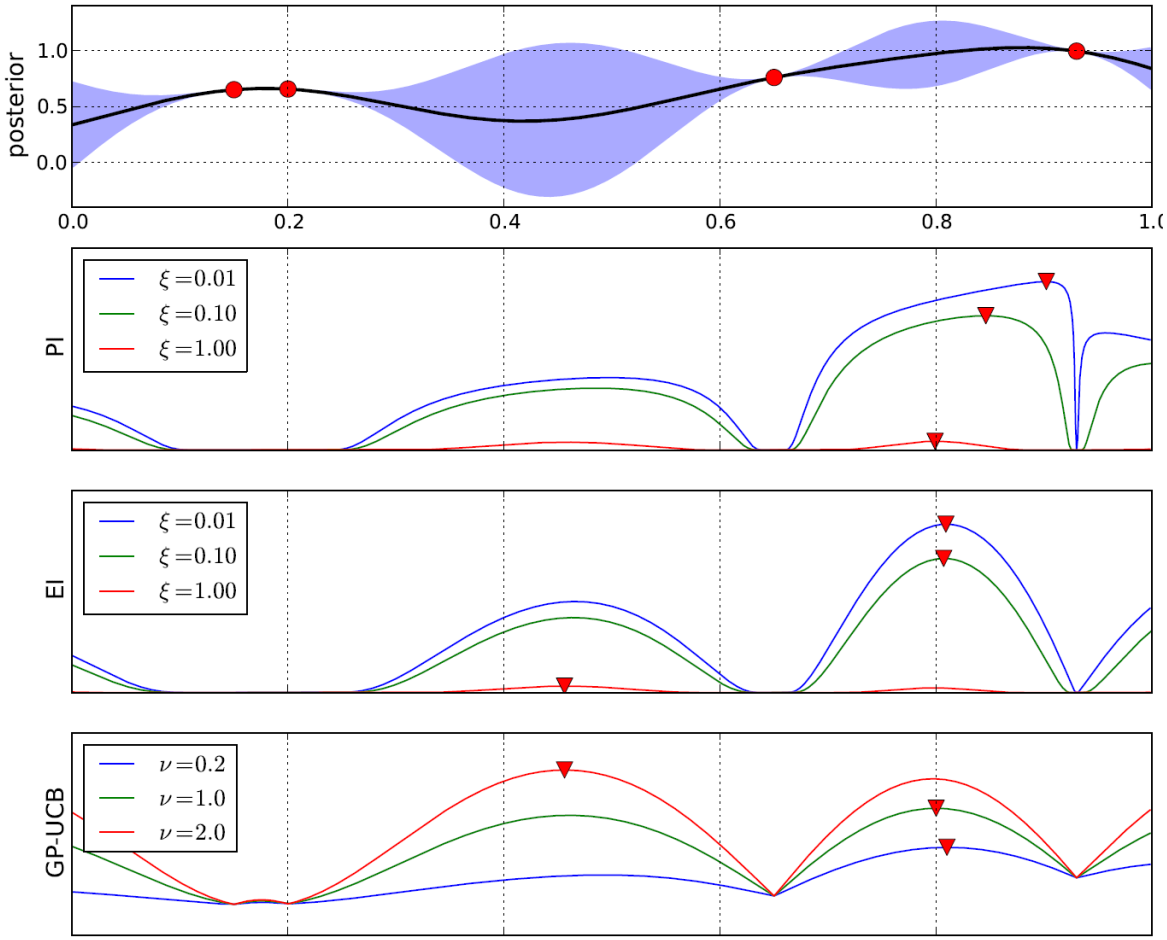
Upper Confidence Bounds (UCB)

$$\mathrm{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \kappa\sigma(x)$$

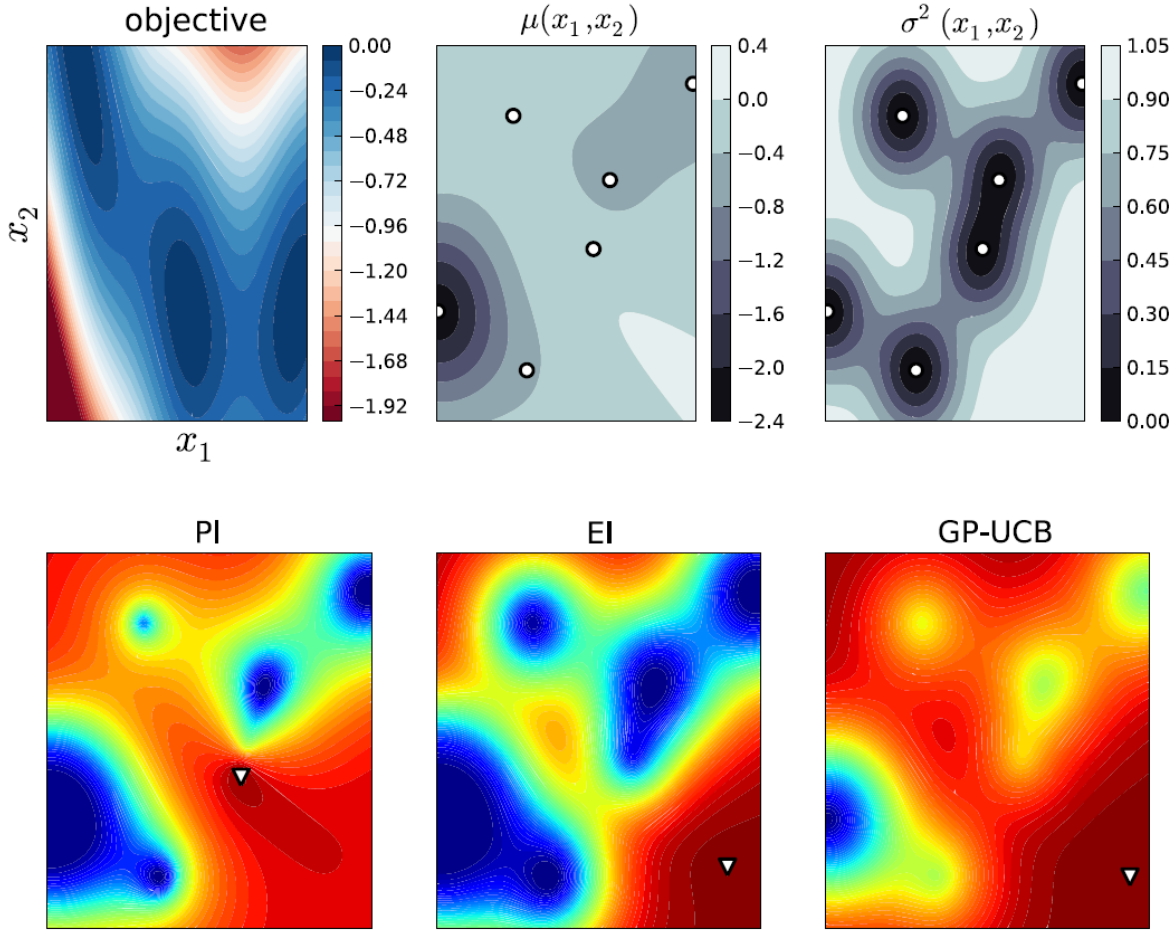
where κ is given

$$\text{GP-UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \sqrt{\nu\tau_t}\sigma(\mathbf{x})$$

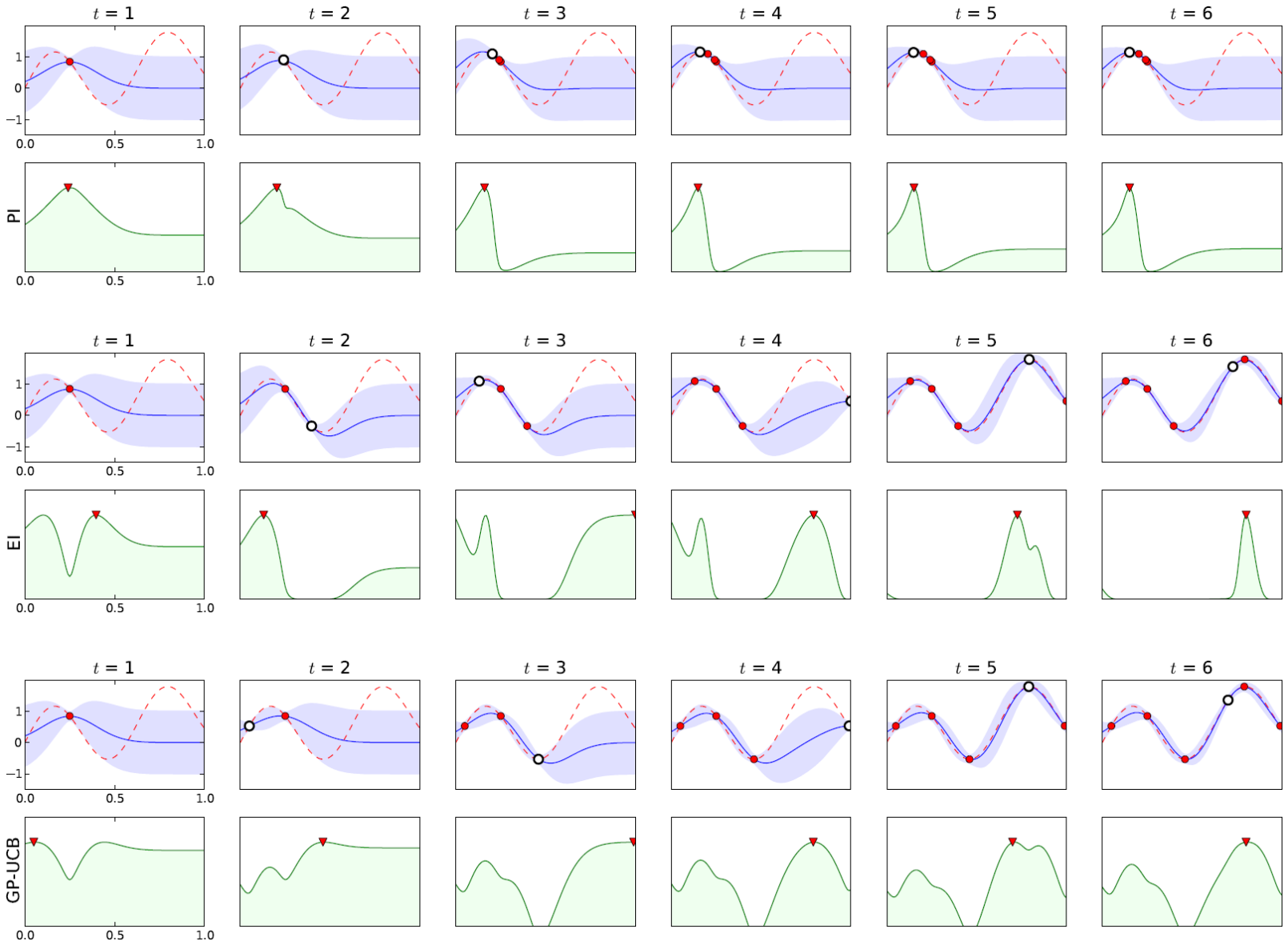
PI vs. EI vs. UCB



Acquisition Functions in 2D

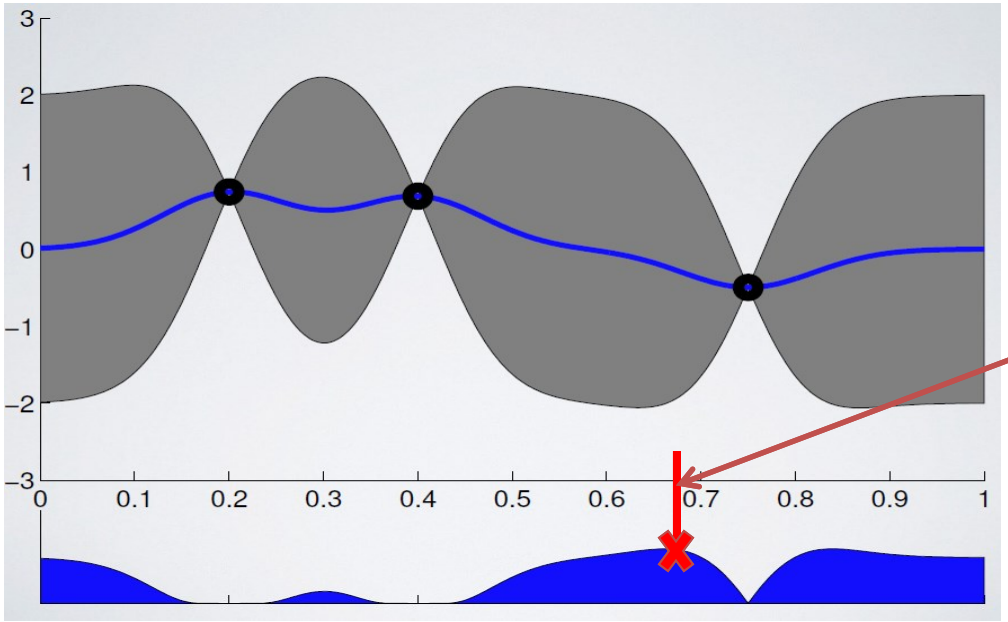


Comparison



Maximizing the Acquisition Function

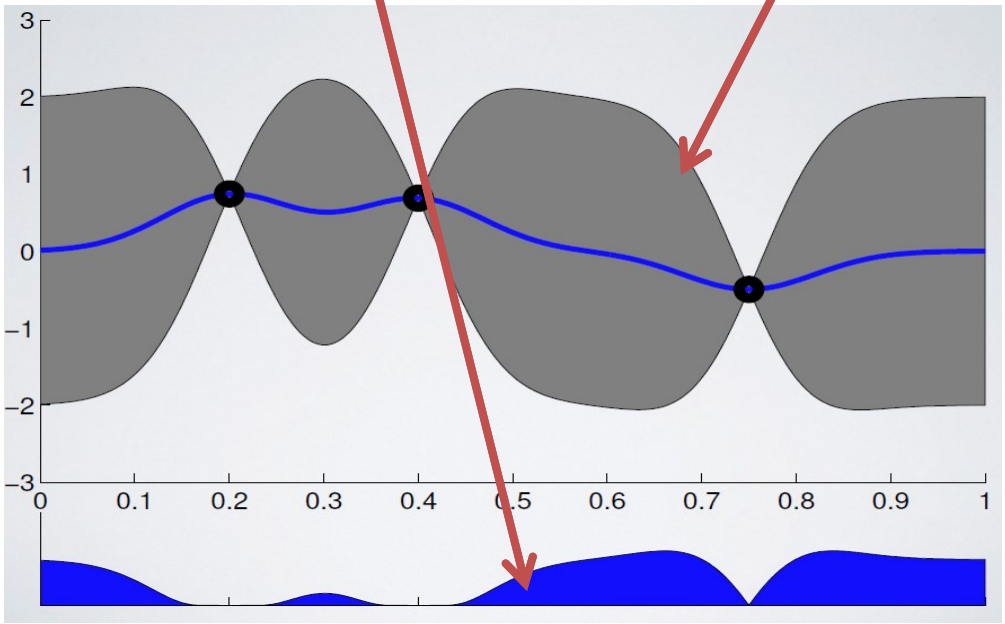
- Unlike the original unknown objective function, it can be cheaply sampled.
- Optimize using DIRECT (DIvide feasible space into fine RECTangles)



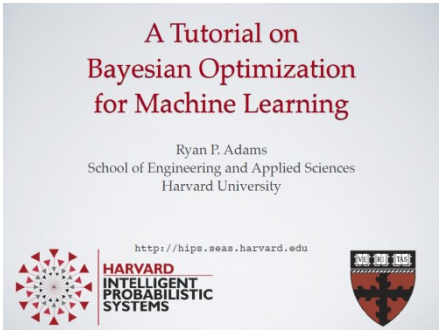
Bayesian Optimization

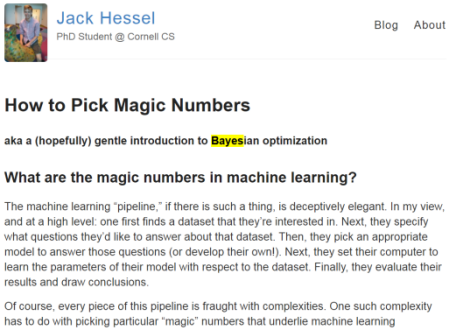
Algorithm 1 Bayesian Optimization

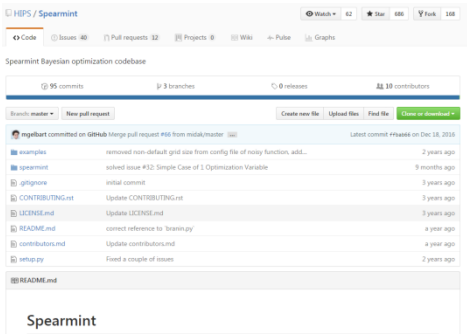
1: **for** $t = 1, 2, \ldots$ **do**
2: Find $\mathbf{x}_t$ by optimizing the acquisition function over the GP: $\mathbf{x}_t = \text{argmax}_{\mathbf{x}}\, u(\mathbf{x}|\mathcal{D}_{1:t-1})$.
3: Sample the objective function: $y_t = f(\mathbf{x}_t) + \varepsilon_t$.
4: Augment the data $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (\mathbf{x}_t, y_t)\}$ and update the GP.
5: **end for**

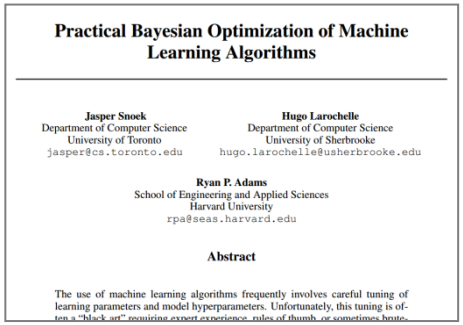


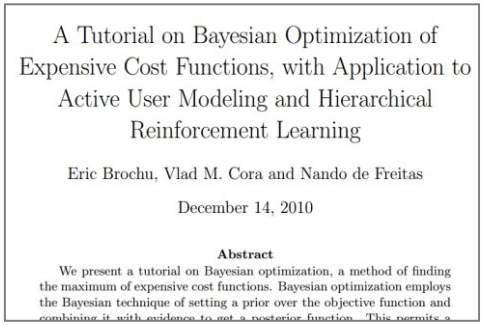
References


A Tutorial on
Bayesian Optimization
for Machine Learning
Ryan P. Adams
School of Engineering and Applied Sciences
Harvard University
http://hips.seas.harvard.edu
HARVARD
INTELLIGENT
PROBABILISTIC
SYSTEMS


Jack Hessel
PhD Student @ Cornell CS
Blog
About
How to Pick Magic Numbers
aka a (hopefully) gentle introduction to Bayesian optimization
What are the magic numbers in machine learning?
The machine learning "pipeline," if there is such a thing, is deceptively elegant. In my view, and at a high level: one first finds a dataset that they're interested in. Next, they specify what questions they'd like to answer about that dataset. Then, they pick an appropriate model to answer those questions (or develop their own!). Next, they set their computer to learn the parameters of their model with respect to the dataset. Finally, they evaluate their results and draw conclusions.
Of course, every piece of this pipeline is fraught with complexities. One such complexity has to do with picking particular "magic" numbers that underlie machine learning


HIPS / Spearmint
Spearmint Bayesian optimization codebase
Spearmint


Practical Bayesian Optimization of Machine Learning Algorithms
Jasper Snoek
Department of Computer Science
University of Toronto
jasper@cs.toronto.edu
Hugo Larochelle
Department of Computer Science
University of Sherbrooke
hugo.larochelle@usherbrooke.edu
Ryan P. Adams
School of Engineering and Applied Sciences
Harvard University
rpa@seas.harvard.edu
Abstract
The use of machine learning algorithms frequently involves careful tuning of learning parameters and model hyperparameters. Unfortunately, this tuning is of-


A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning
Eric Brochu, Vlad M. Cora and Nando de Freitas
December 14, 2010
Abstract
We present a tutorial on Bayesian optimization, a method of finding the maximum of expensive cost functions. Bayesian optimization employs the Bayesian technique of setting a prior over the objective function and


Machine learning - Bayesian optimization and multi-armed bandits
Nando de Freitas
Subscribe
23,623

Thank You