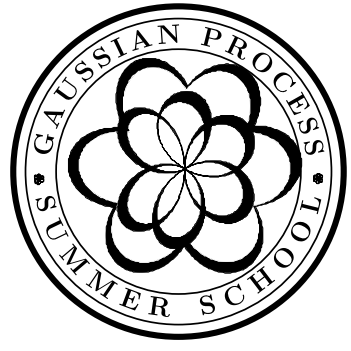
Introduction to Bayesian Optimization

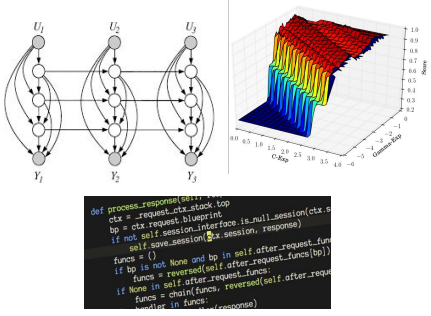
Javier González

Masterclass, 7-February, 2107 @Lancaster University



Data Science pipeline/Autonomous System

Challenges and needs for automation



Data Collection

Optimal design

Features extraction

Filters, dimensionality reduction

Modelling

Model tuning and configuration, code optimisation

Results Interpretation/ Decision

Data visualisation, Sequential experimentation

Pipeline improvement / Interaction with environment

Experimental Design - Uncertainty Quantification

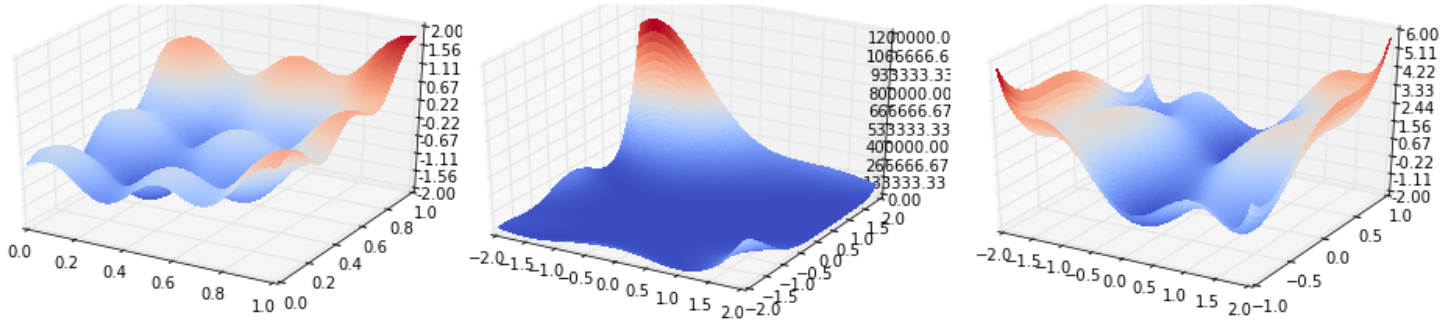
Can we automate/simplify the process of designing complex experiments?

Emulator - Simulator - Physical system

Global optimization

Consider a 'well behaved' function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^D$ is a bounded domain.

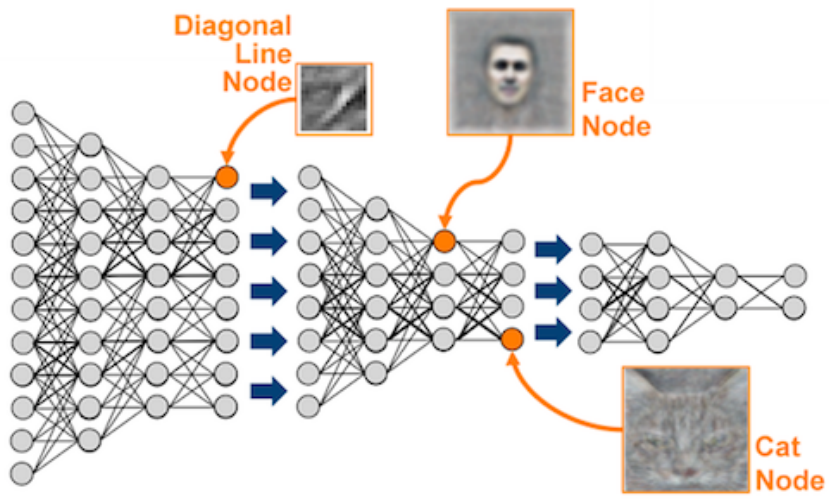
$$x_M = \arg\min_{x \in \mathcal{X}} f(x).$$



- f is explicitly unknown and multimodal.
- Evaluations of f may be perturbed.
- Evaluations of f are expensive.

Expensive functions, who doesn't have one?

Parameter tuning in ML algorithms.



- Number of layers/units per layer
- Weight penalties
- Learning rates, etc.

Figure source: http://theanalyticsstore.com/deep-learning

Expensive functions, who doesn't have one?

Many other problems:

- Robotics, control, reinforcement learning.
- Scheduling, planning
- compilers, hardware, software?
- Intractable likelihoods.

What to do?

Option 1: Use previous knowledge

To select the parameters at hand. Perhaps not very scientific but still in use...

What to do?

Option 2: Grid search?

If f is L-Lipschitz continuous and we are in a noise-free domain to guarantee that we propose some $x_{M,n}$ such that

$$f(x_M) - f(x_{M,n}) \leq \epsilon$$

we need to evaluate f on a D-dimensional unit hypercube:

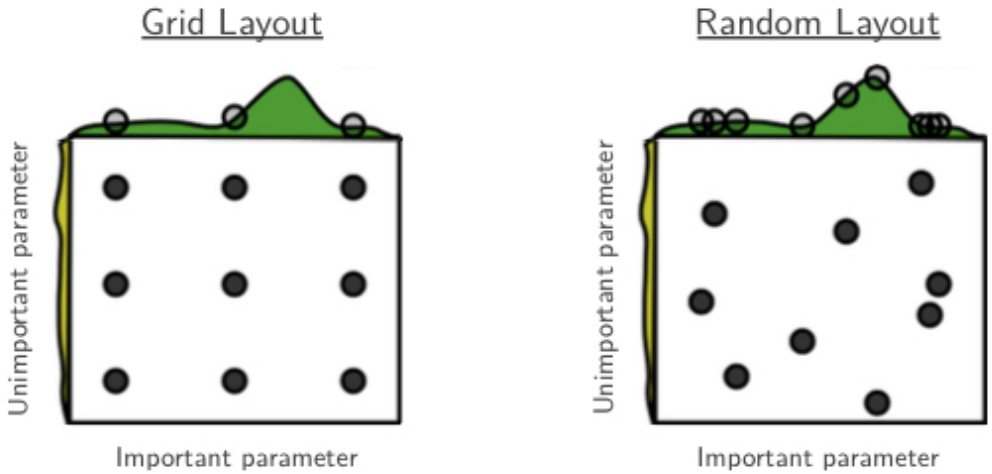
$$(L/\epsilon)^D evaluations!$$

Example: $(10/0.01)^5 = 10e14...$
... but function evaluations are very expensive!

What to do?

Option 3: Random search?

We can sample the space uniformly [Bergstra and Bengio 2012]



Better than grid search in various senses but still expensive to guarantee good coverage.

What to do?

Key question:

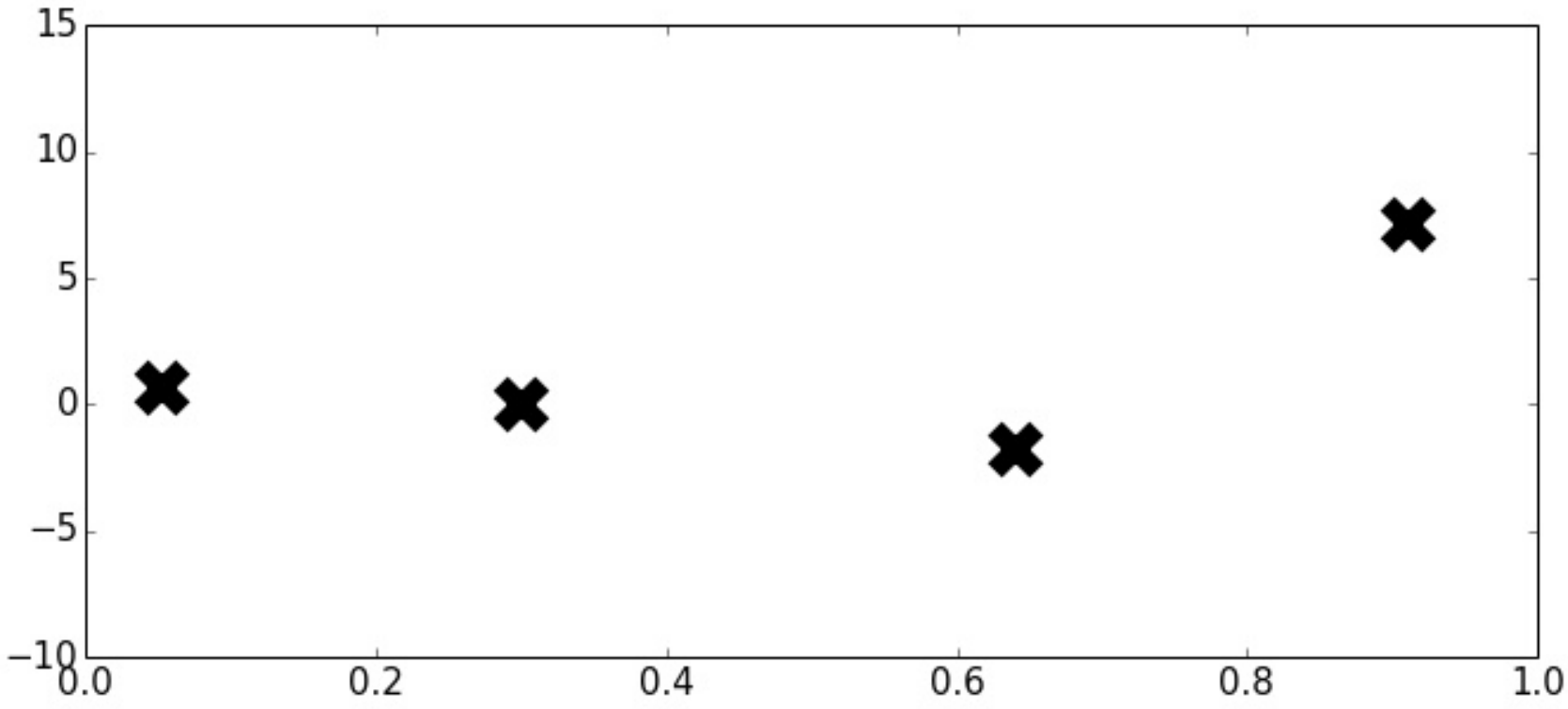
Can we do better?

Problem (the audience is encouraged to participate!)

- Find the optimum of some function f in the interval [0,1].
- f is L-Lipchitz continuous and differentiable.
- Evaluations of f are exact and we have 4 of them!

Situation

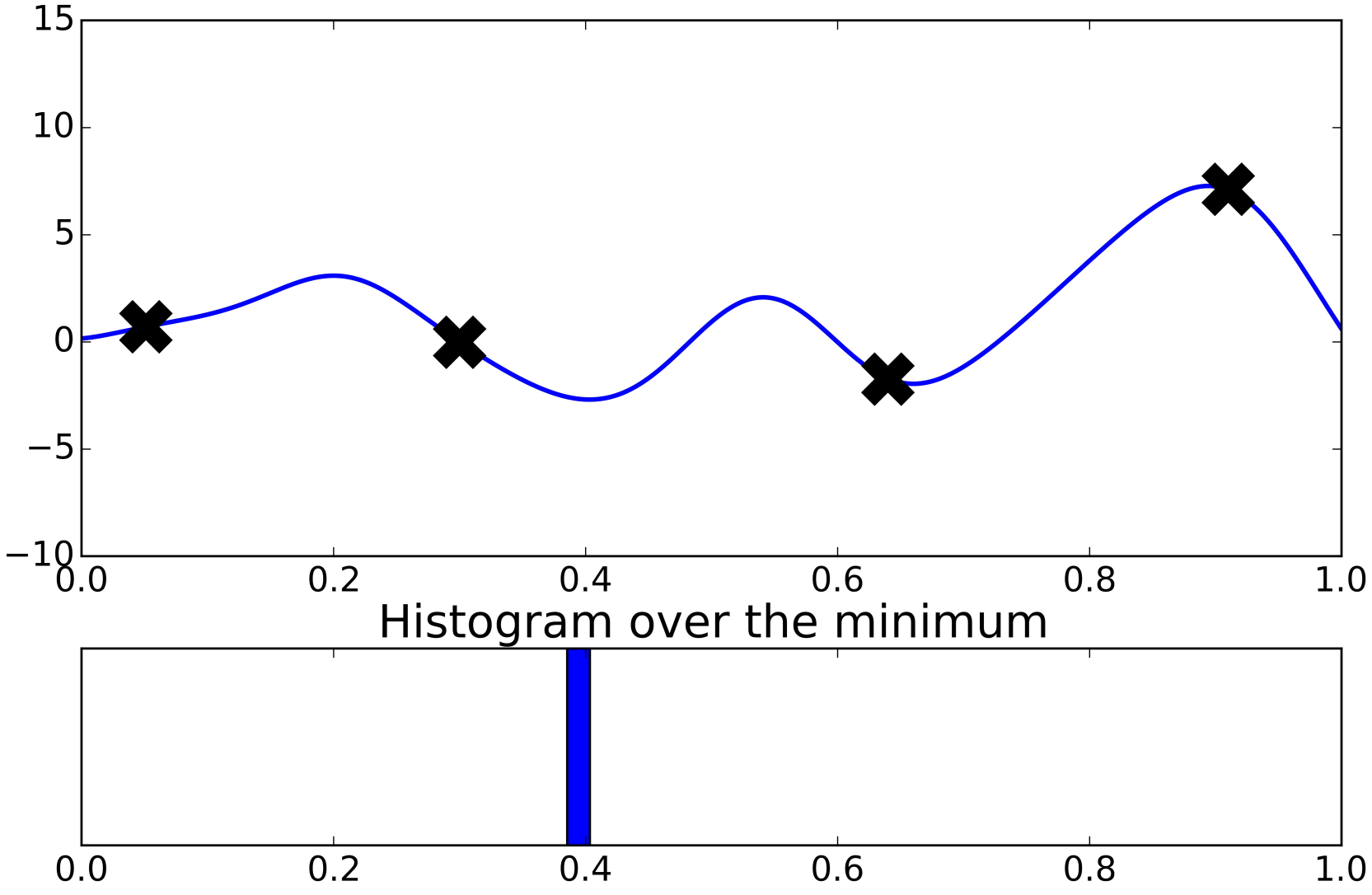
We have a few function evaluations



Where is the minimum of f?
Where should the take the next evaluation?

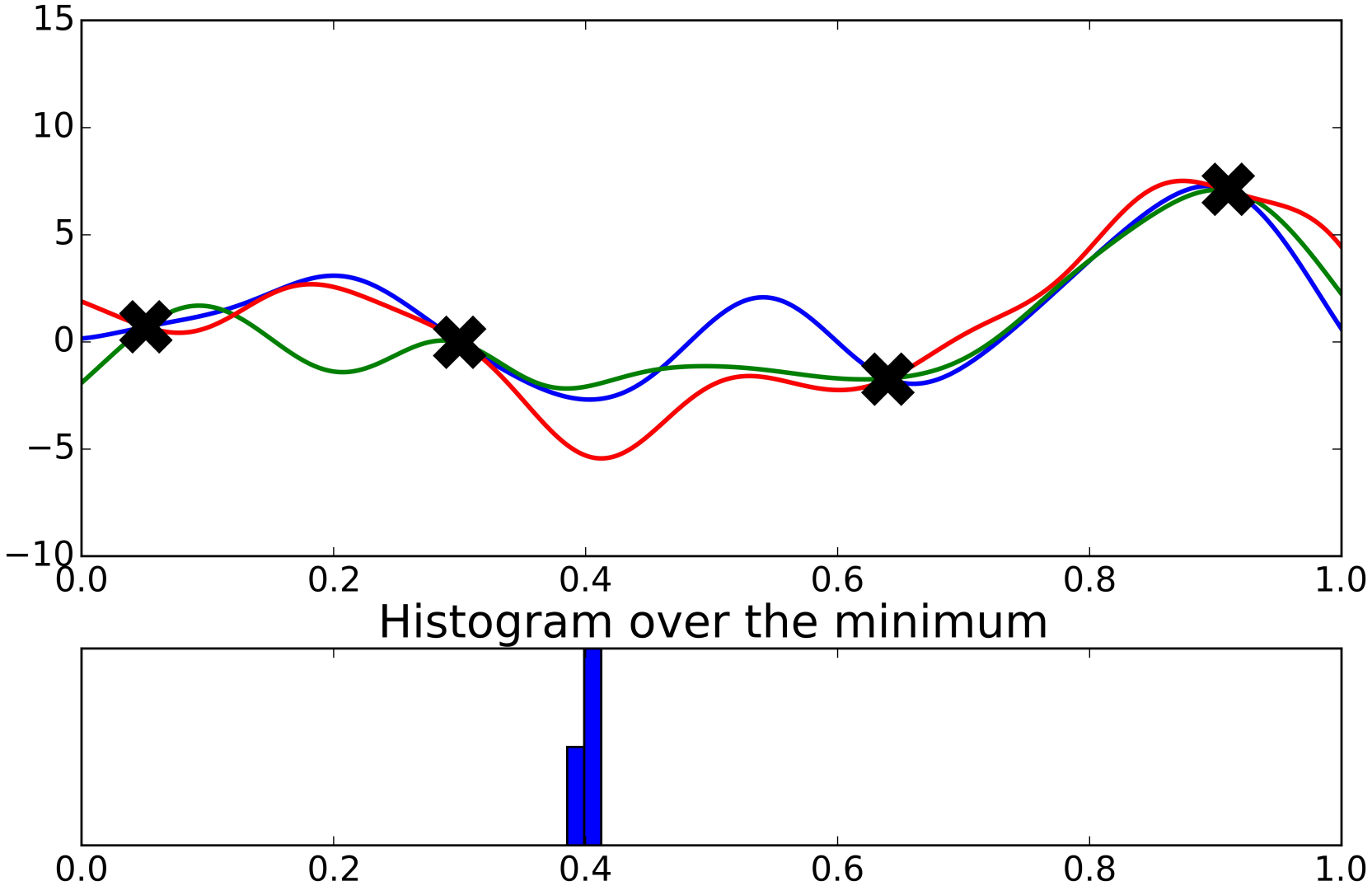
Intuitive solution

One curve



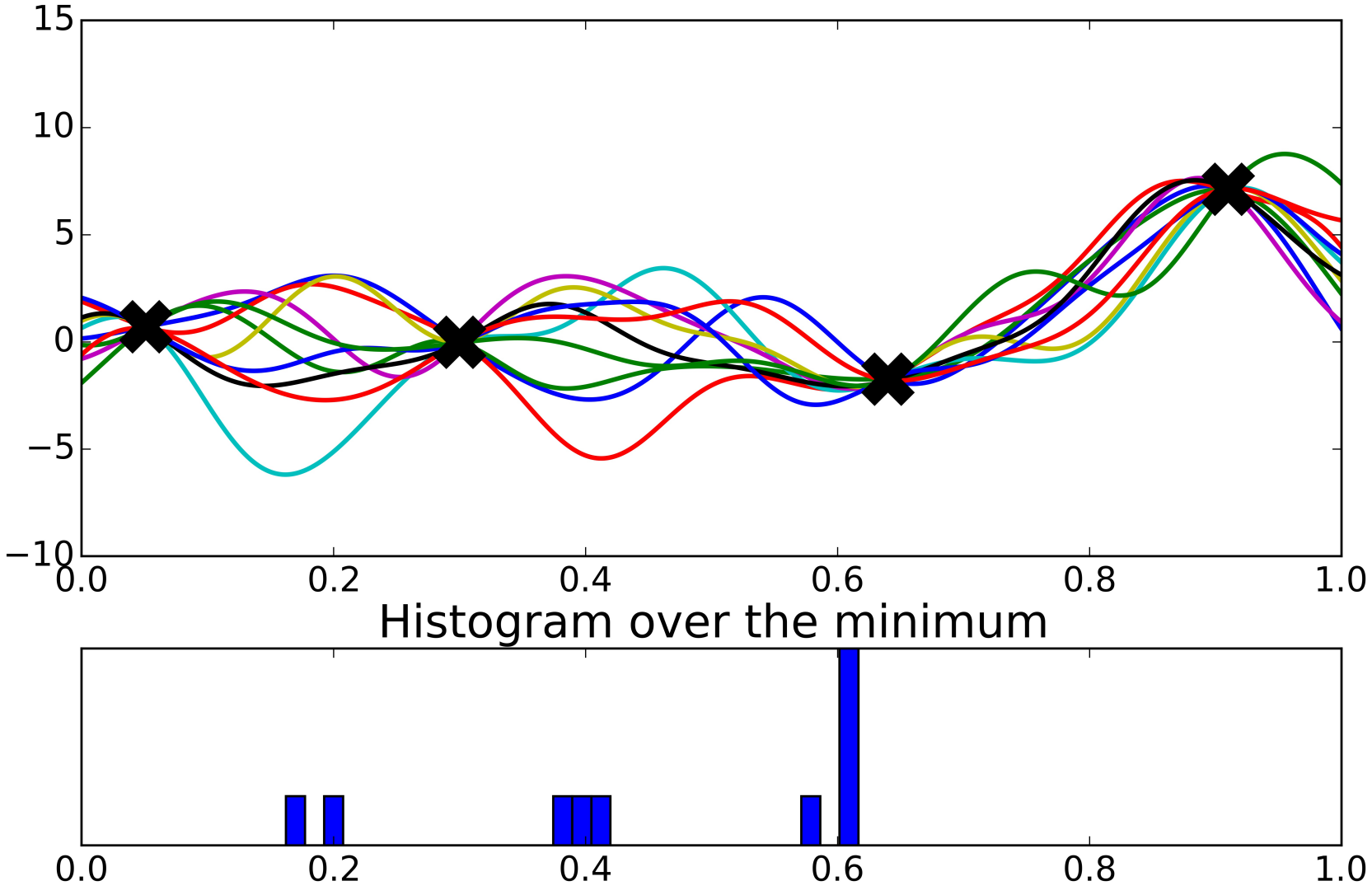
Intuitive solution

Three curves



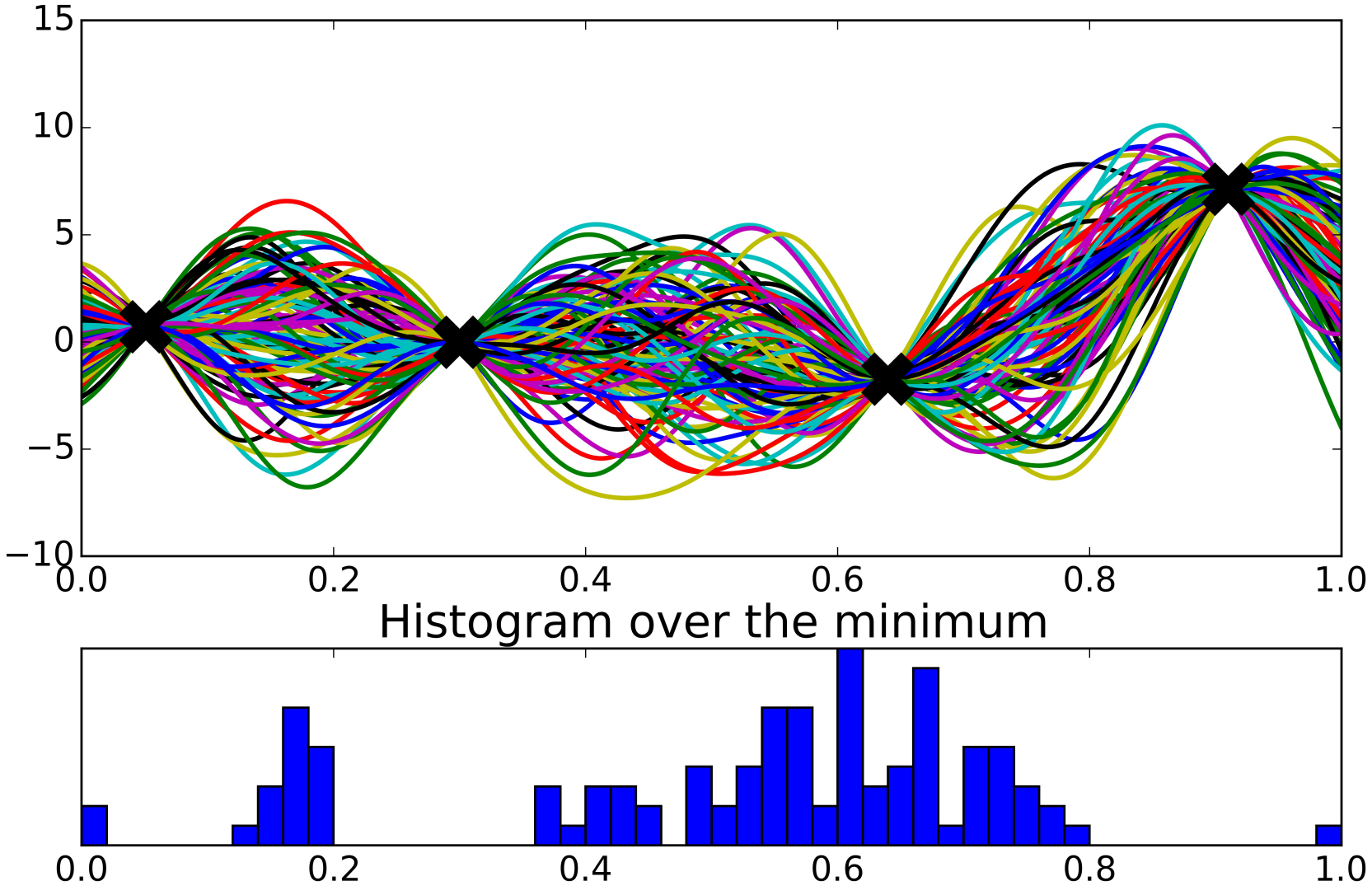
Intuitive solution

Ten curves



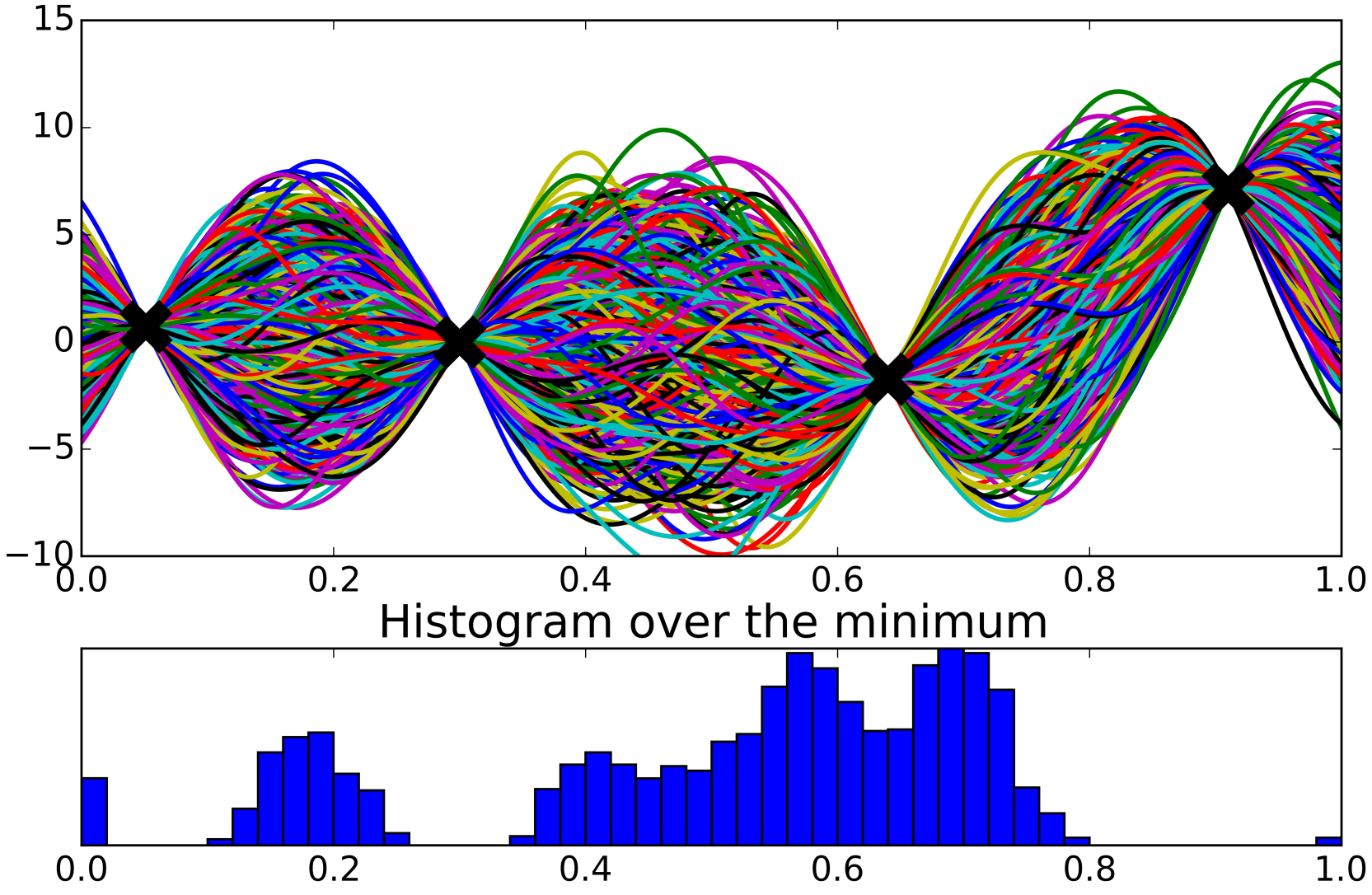
Intuitive solution

Hundred curves



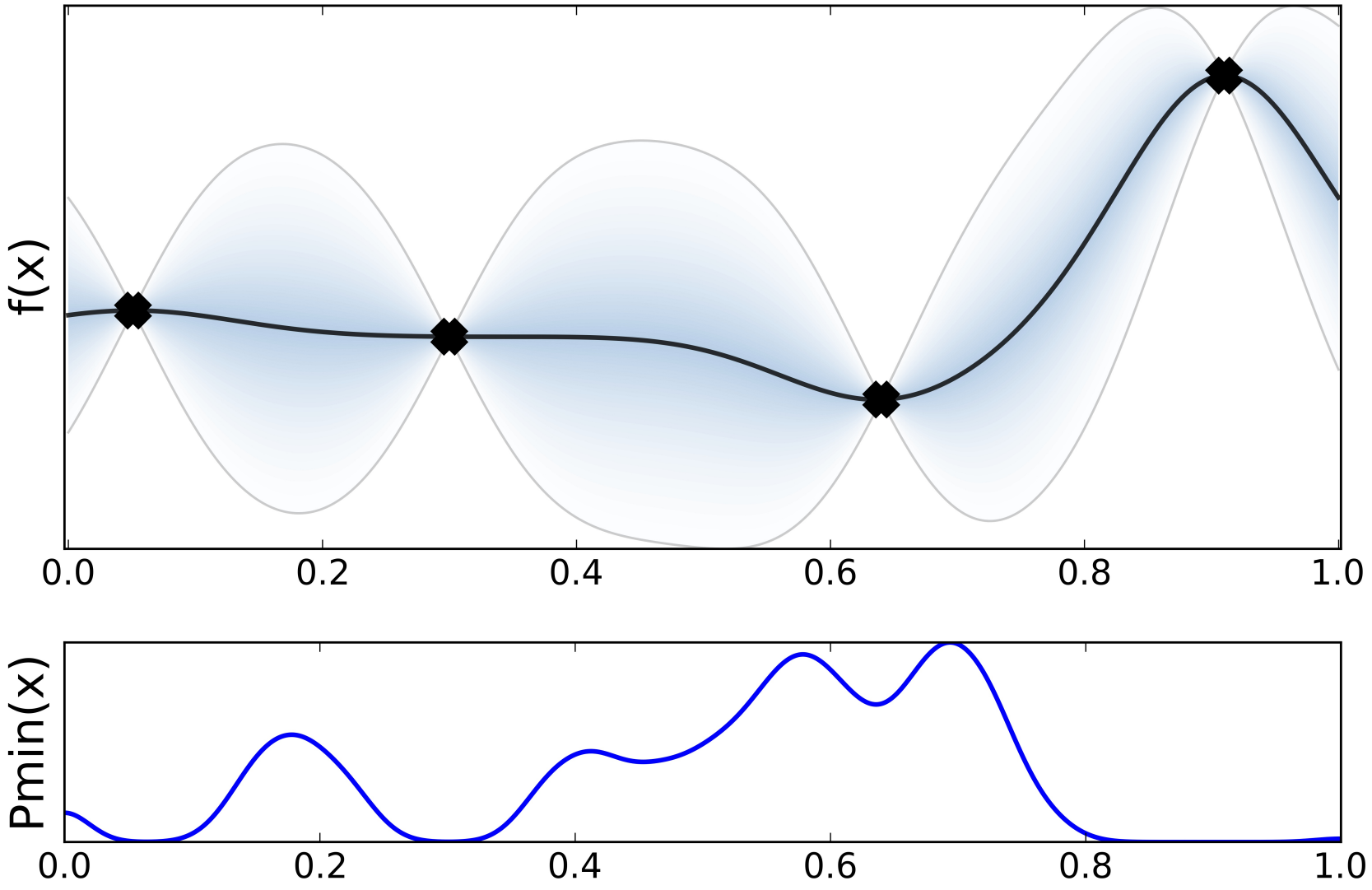
Intuitive solution

Many curves



Intuitive solution

Infinite curves



General idea: surrogate modelling

1. Use a surrogate model of f to carry out the optimization.

2. Define an utility function to collect new data points satisfying some optimality criterion: *optimization* as *decision*.

3. Study *decision* problems as *inference* using the surrogate model: use a probabilistic model able to calibrate both, epistemic and aleatoric uncertainty.

Uncertainty Quantification

Bayesian Optimisation

[Mockus, 1978]

Methodology to perform global optimisation of multimodal black-box functions.

1. Choose some *prior measure* over the space of possible objectives f.

2. Combine prior and the likelihood to get a *posterior measure* over the objective given some observations.

3. Use the posterior to decide where to take the next evaluation according to some *acquisition/loss function*.

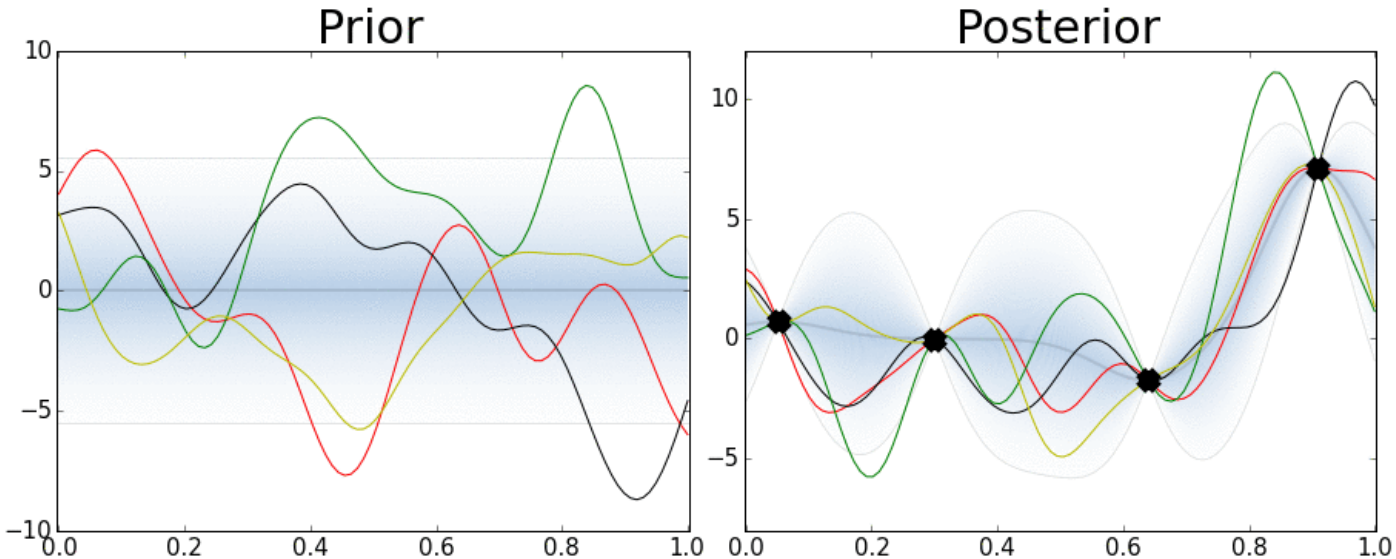
4. Augment the data.

Iterate between 2 and 4 until the evaluation budget is over.

Surrogate model: Gaussian process

Default Choice: Gaussian processes [Rasmunsen and Williams, 2006]

Infinite-dimensional probability density, such that each linear finite-dimensional restriction is multivariate Gaussian.



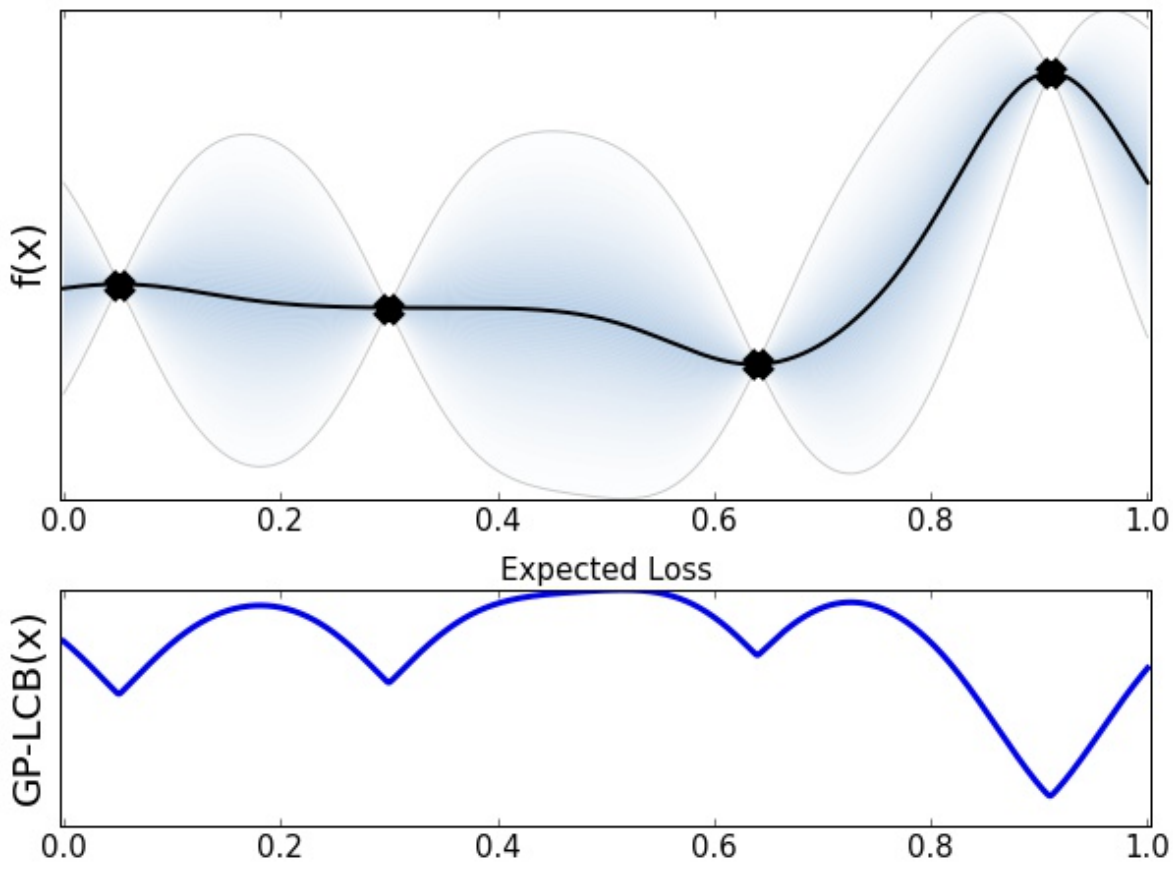
- Model $f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$ is determined by the *mean function* $m(x)$ and *covariance function* $k(x, x'; \theta)$.
- Posterior mean $\mu(x; \theta, \mathcal{D})$ and variance $\sigma(x; \theta, \mathcal{D})$ can be *computed explicitly* given a dataset $\mathcal{D}$.

GP Upper (lower) Confidence Band

[Srinivas et al., 2010]

Direct balance between exploration and exploitation:

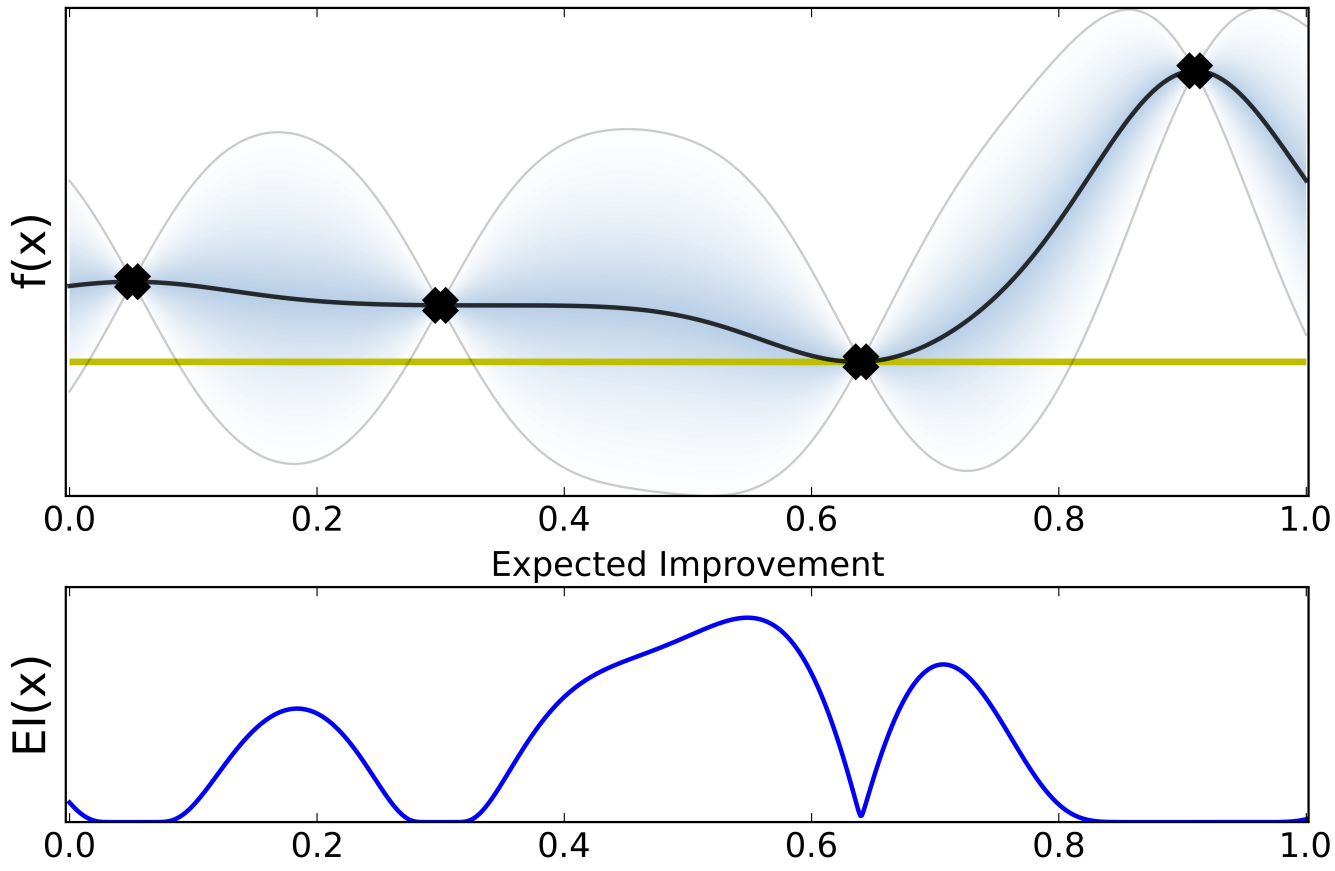
$$\alpha_{LCB}(\mathbf{x}; \theta, \mathcal{D}) = -\mu(\mathbf{x}; \theta, \mathcal{D}) + \beta_t \sigma(\mathbf{x}; \theta, \mathcal{D})$$



Expected Improvement

[Jones et al., 1998]

$$\alpha_{EI}(\mathbf{x};\theta,\mathcal{D}) = \int_y \max(0, y_{best} - y) p(y|\mathbf{x};\theta,\mathcal{D}) dy$$



Expected Improvement

[Jones et al., 1998]

- Perhaps the most used acquisition.
- Explicit for available for Gaussian posteriors.
- It is too greedy in some problems. It is possible to make more explorative adding a 'explorative' parameter

$$\alpha_{EI}(\mathbf{x}; \theta, \mathcal{D}) = \sigma(\mathbf{x}; \theta, \mathcal{D})(\gamma(x)\Phi(\gamma(x))) + \mathcal{N}(\gamma(x); 0, 1).$$

where

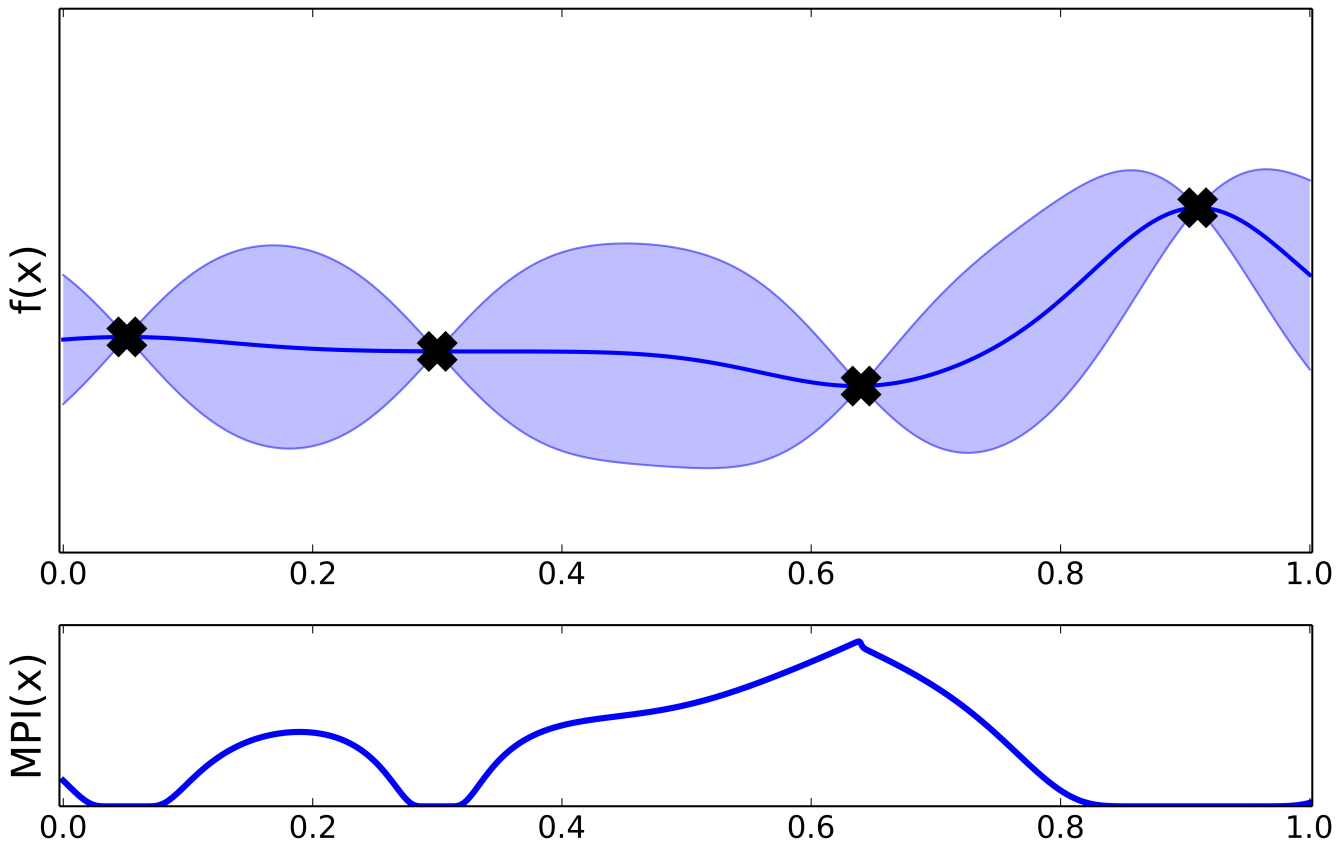
$$\gamma(x) = \frac{f(x_{best}) - \mu(\mathbf{x}; \theta, \mathcal{D}) + \psi}{\sigma(\mathbf{x}; \theta, \mathcal{D})}.$$

Maximum Probability of Improvement

[Hushner, 1964]

$$\gamma(\mathbf{x}) = \sigma(\mathbf{x}; \theta, \mathcal{D})^{-1}(\mu(\mathbf{x}; \theta, \mathcal{D}) - y_{best})$$

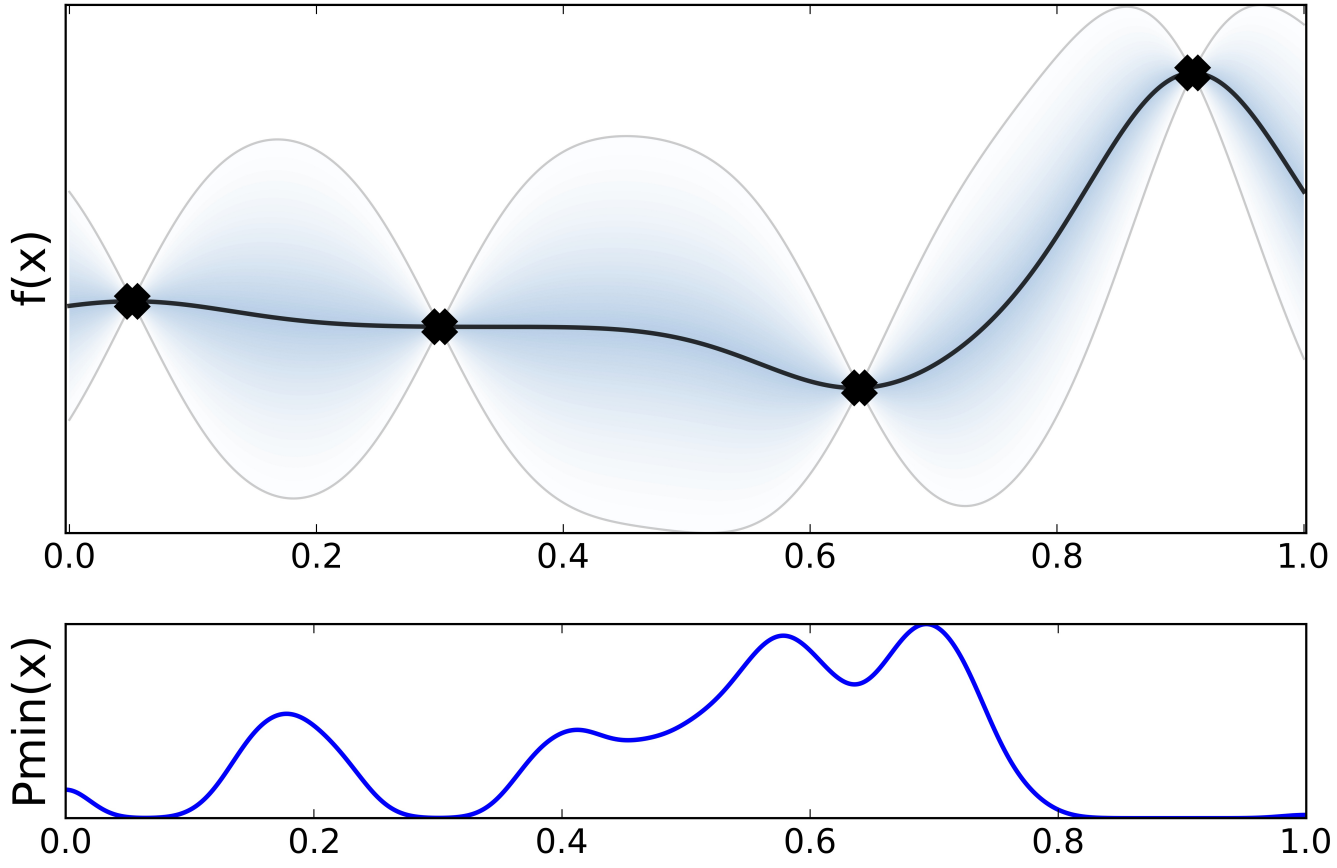
$$\alpha_{MPI}(\mathbf{x}; \theta, \mathcal{D}) = p(f(\mathbf{x}) < y_{best}) = \Phi(\gamma(\mathbf{x}))$$



Information-theoretic approaches

[Hennig and Schuler, 2013; Hernández-Lobato et al., 2014]

$$\alpha_{ES}(\mathbf{x}; \theta, \mathcal{D}) = H[p(x_{min}|\mathcal{D})] - \mathbb{E}_{p(y|\mathcal{D},\mathbf{x})}[H[p(x_{min}|\mathcal{D} \cup \{\mathbf{x}, y\})]]$$



Information-theoretic approaches

Uses the distribution of the minimum

$$p_{min}(x) \equiv p[x = \arg\min f(x)] = \int_{f:I\to\Re} p(f) \prod_{\substack{\tilde{x}\in I \\ \tilde{x}\neq x}} \theta[f(\tilde{x}) - f(x)]df$$

where θ is the Heaviside's step function. No closed form!

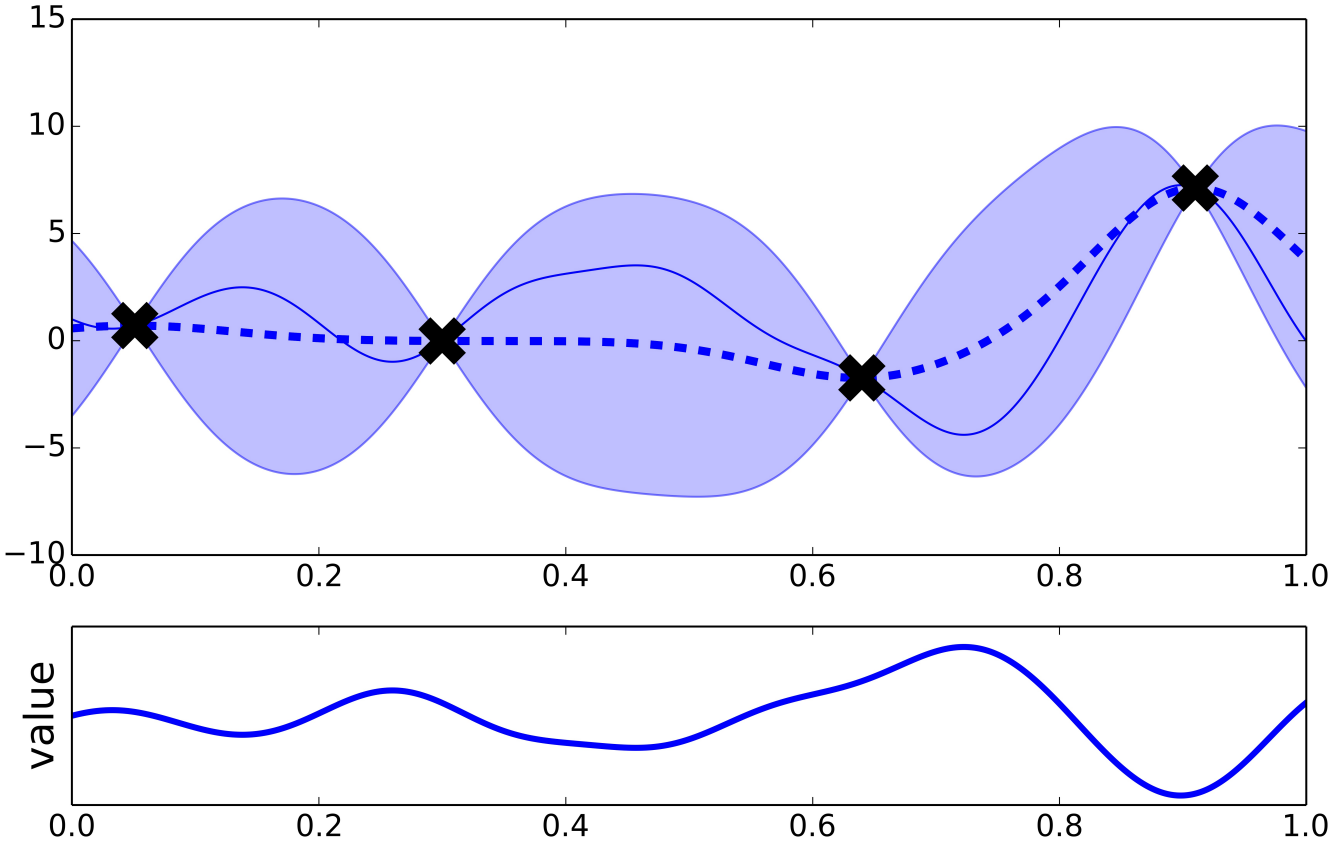
Use Thomson sampling to approximate the distribution. Generate many sample paths from the GP, optimize them to take samples from $p_{min}(x)$.

Thomson sampling

Probability matching

$$\alpha_{THOMSON}(\mathbf{x}; \theta, \mathcal{D}) = g(\mathbf{x})$$

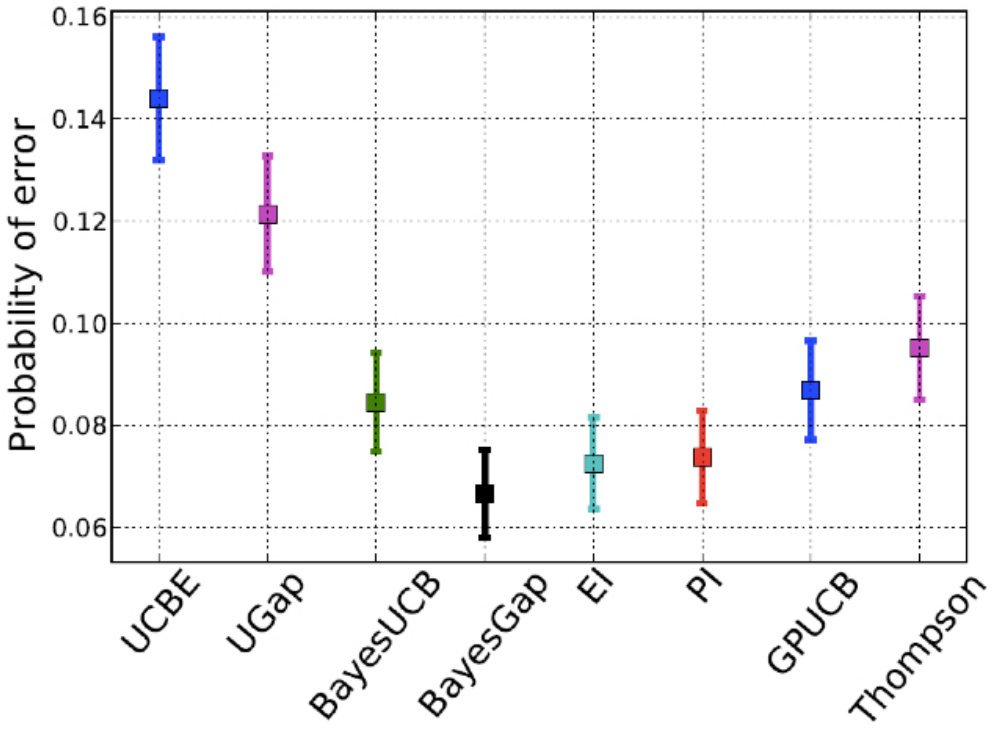
$g(\mathbf{x})$ is sampled form $\mathcal{GP}(\mu(x), k(x, x'))$



The choice of utility matters

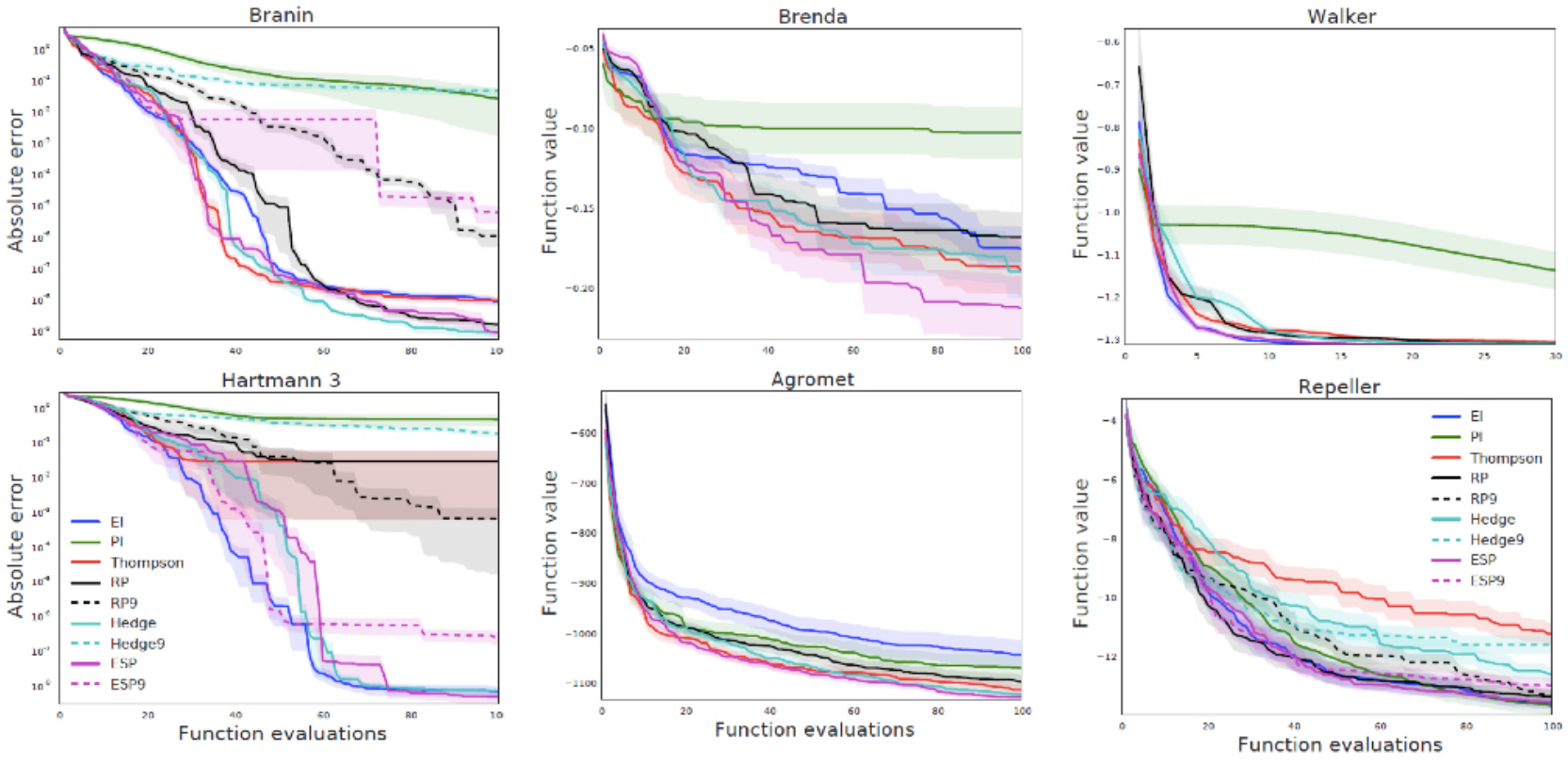
[Hoffman, Shahriari and de Freitas, 2013]

The choice of the utility may change a lot the result of the optimisation.



The choice of utility in practice

[Hoffman, Shahriari and de Freitas, 2013]



The best utility depends on the problem and the level of exploration/exploitation required.

Illustration of BO

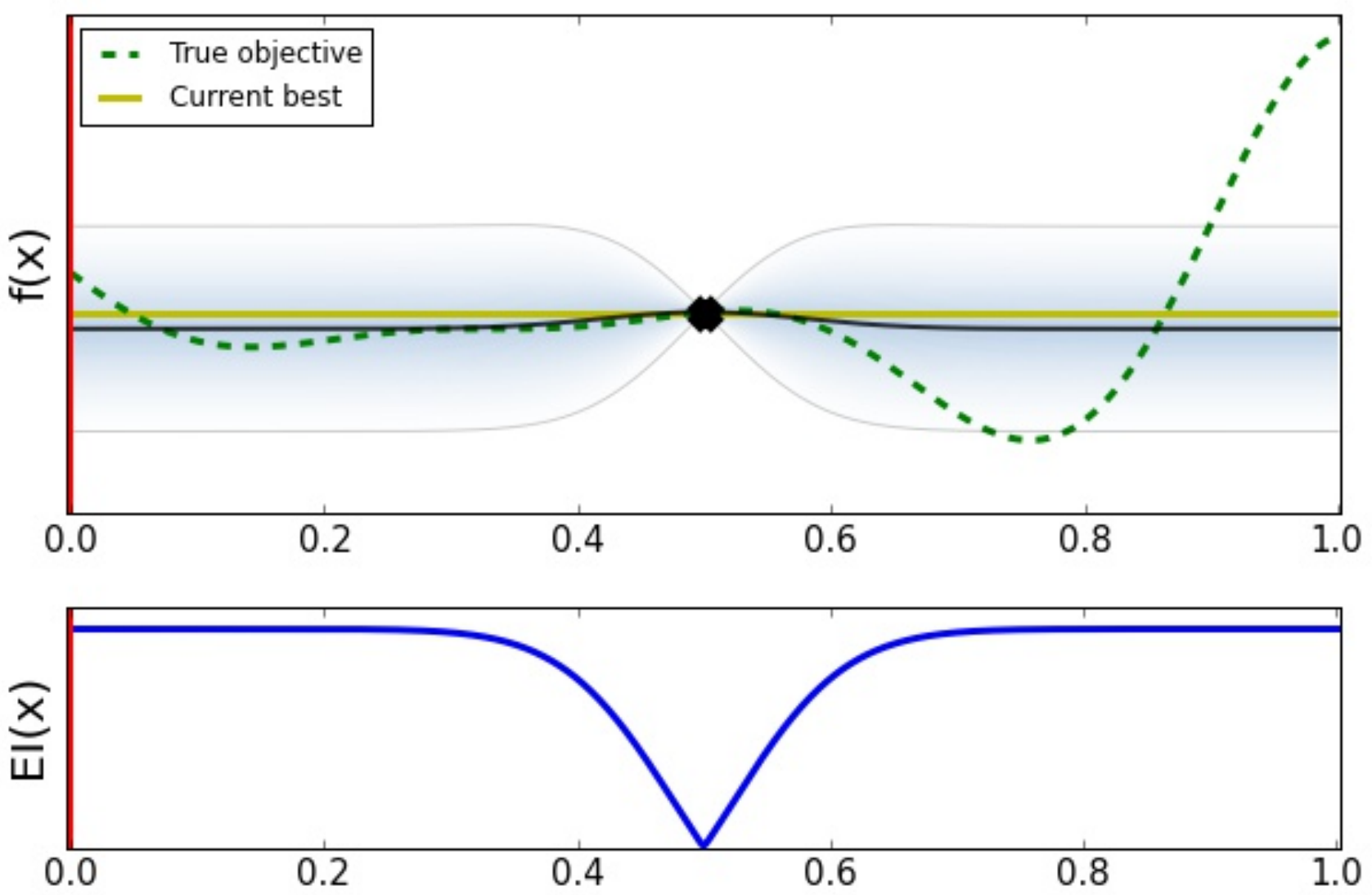


Illustration of BO

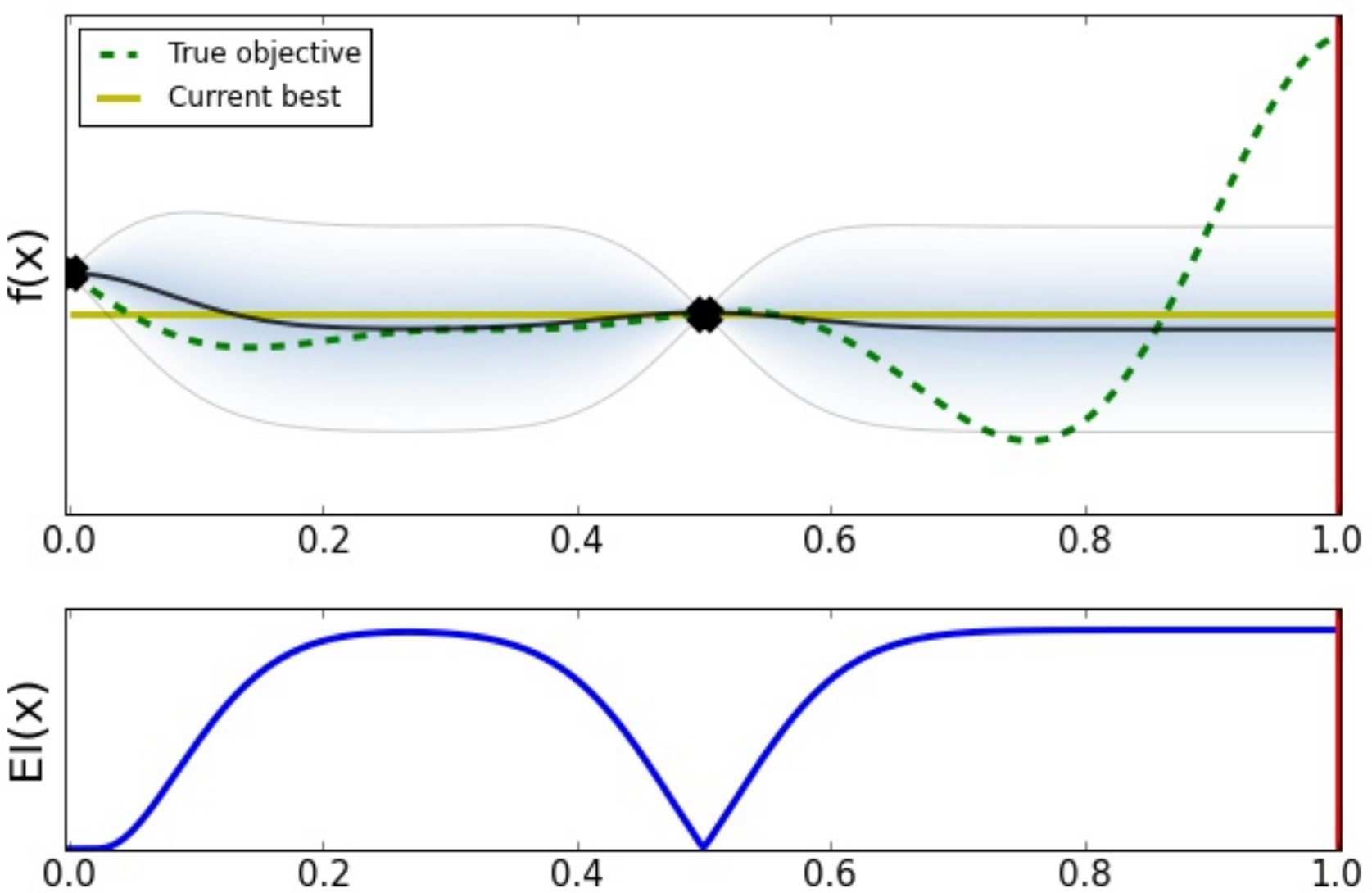


Illustration of BO

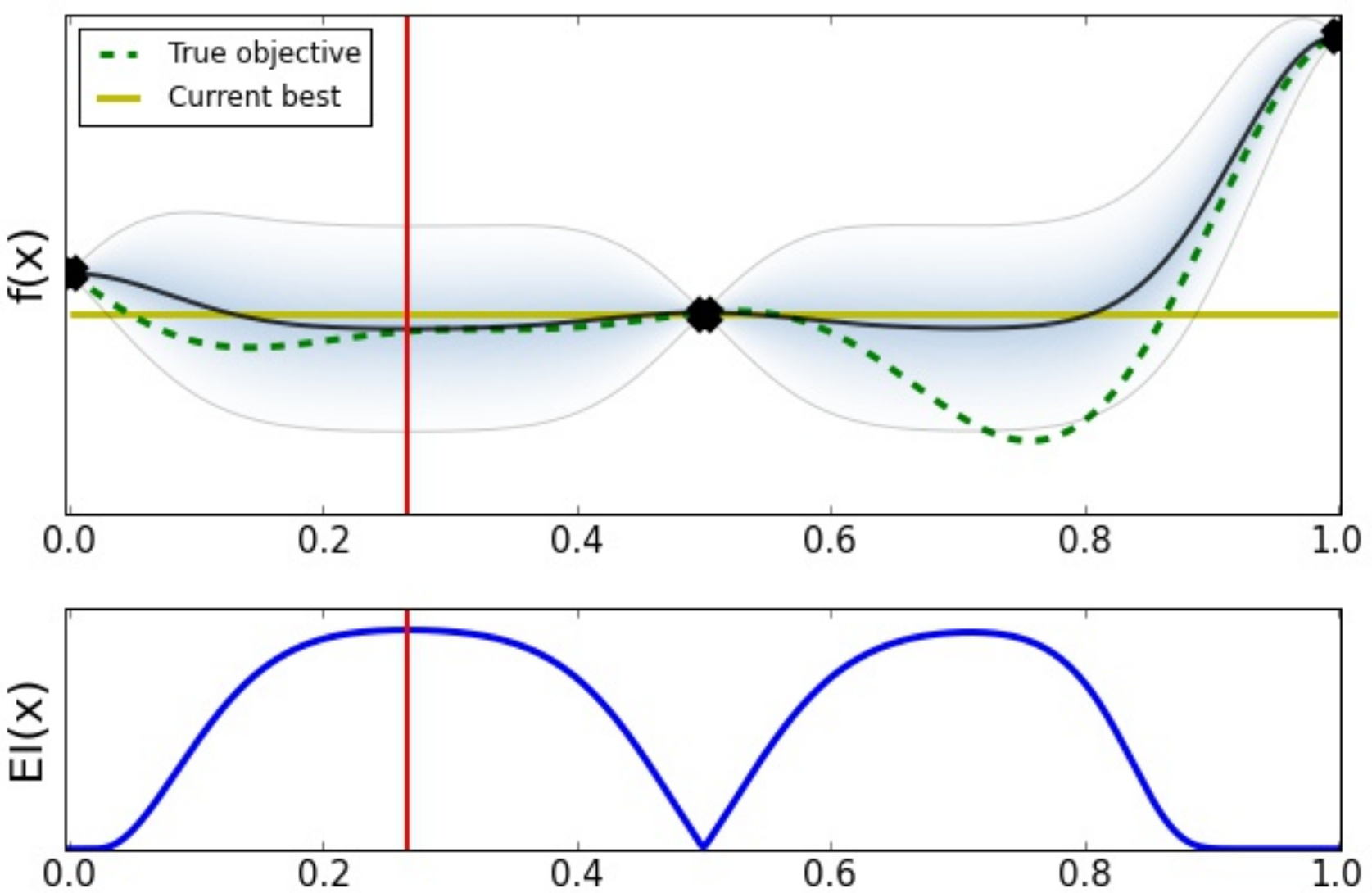


Illustration of BO

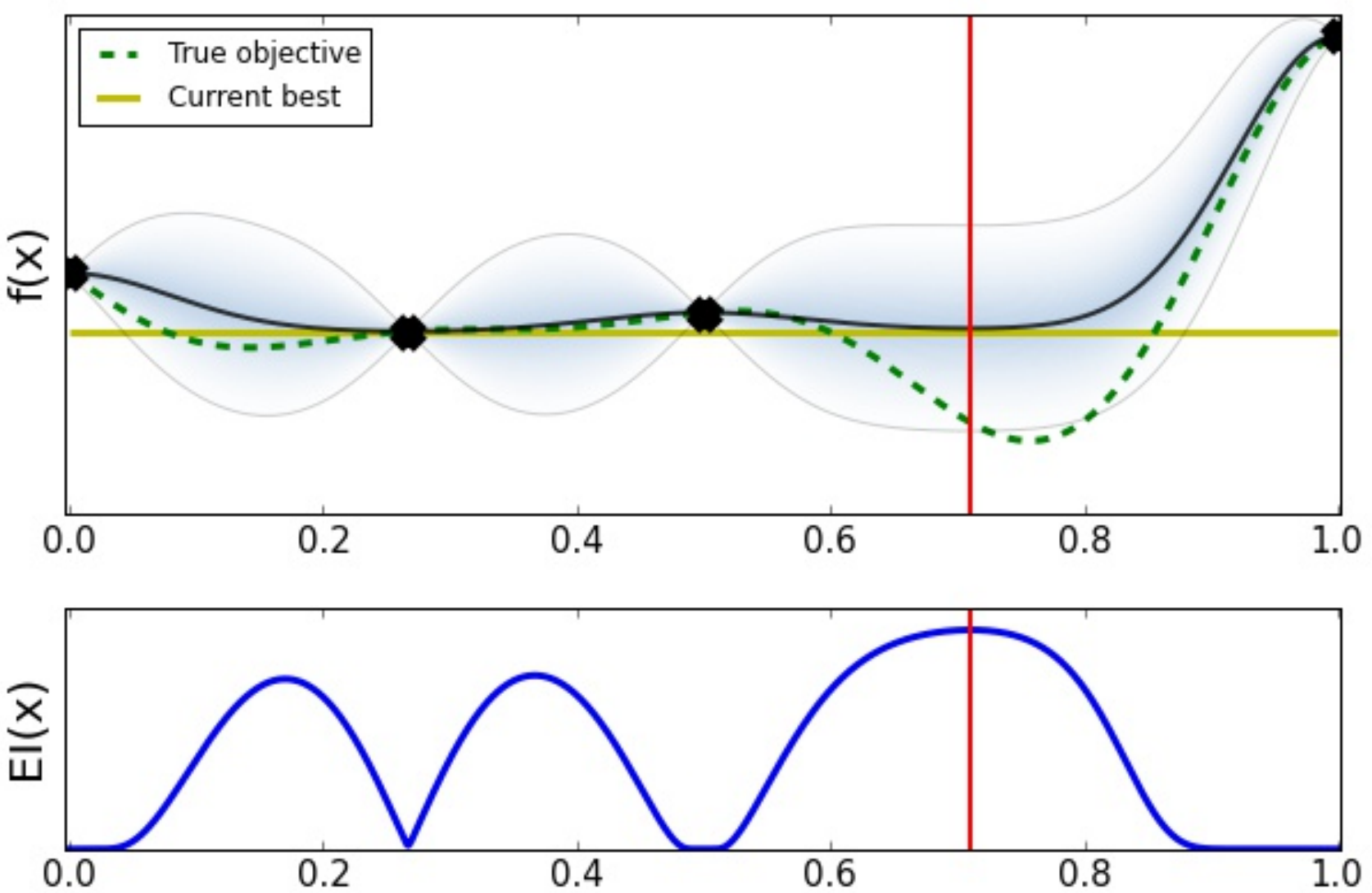


Illustration of BO

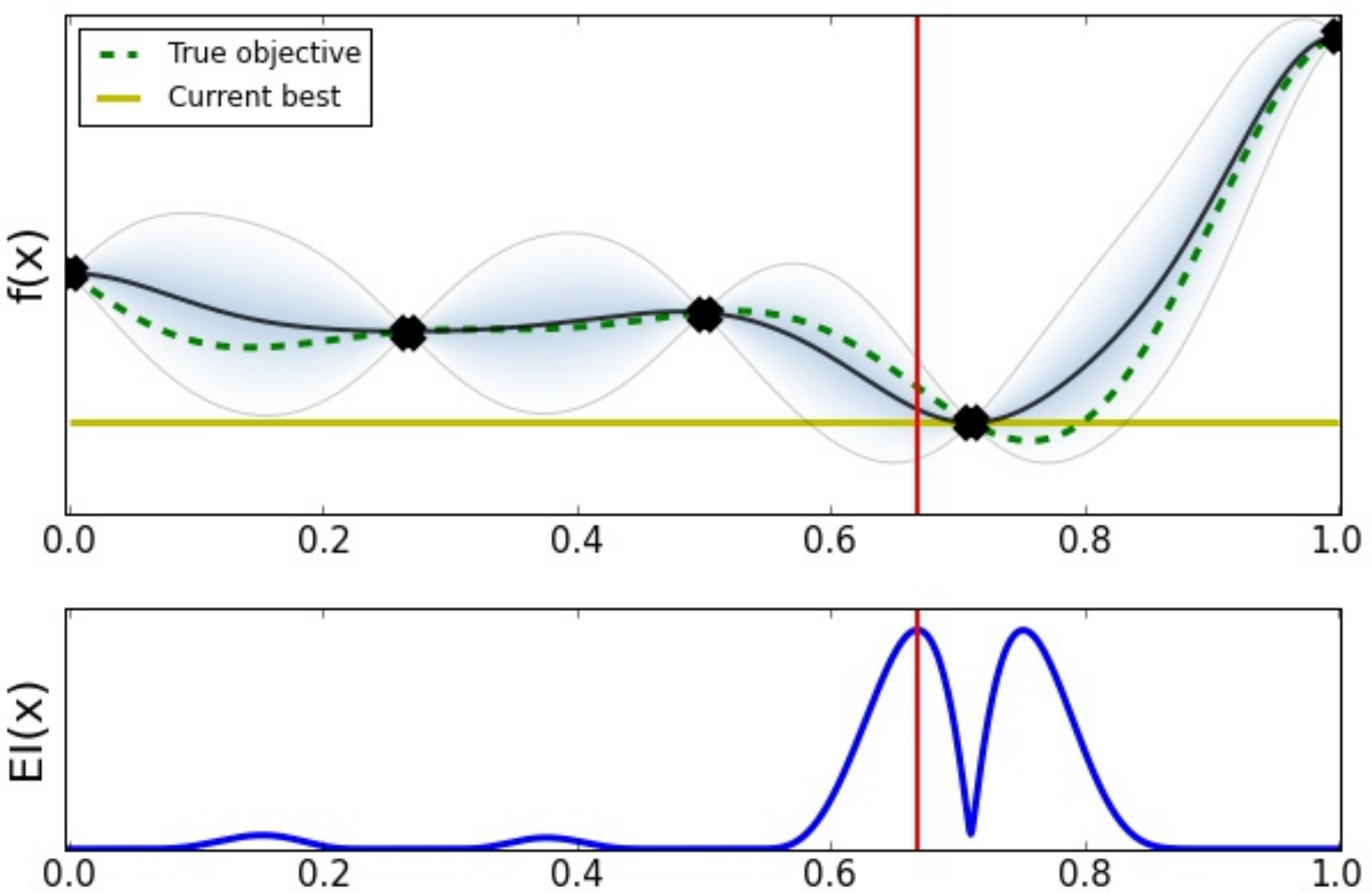


Illustration of BO

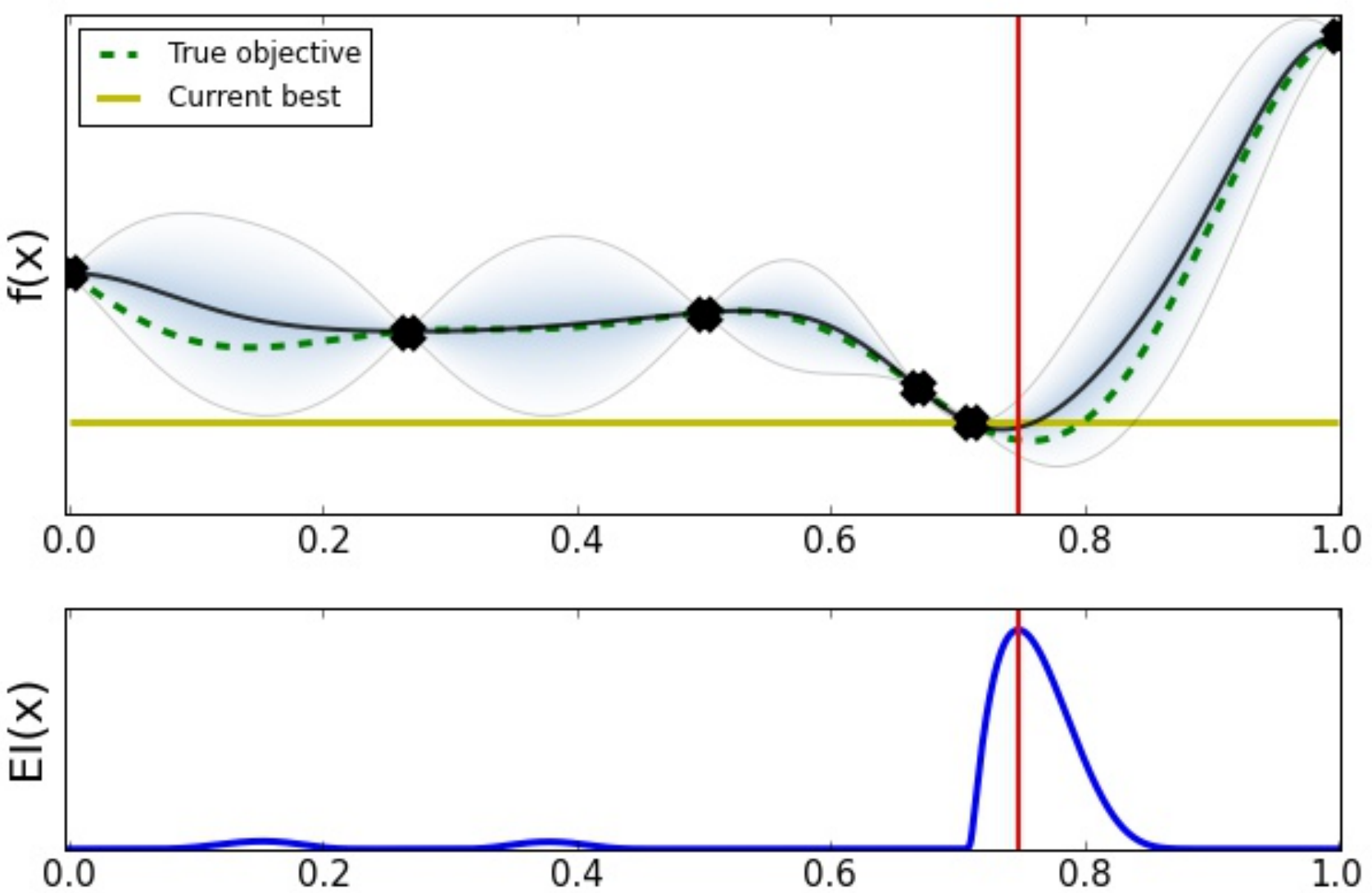


Illustration of BO

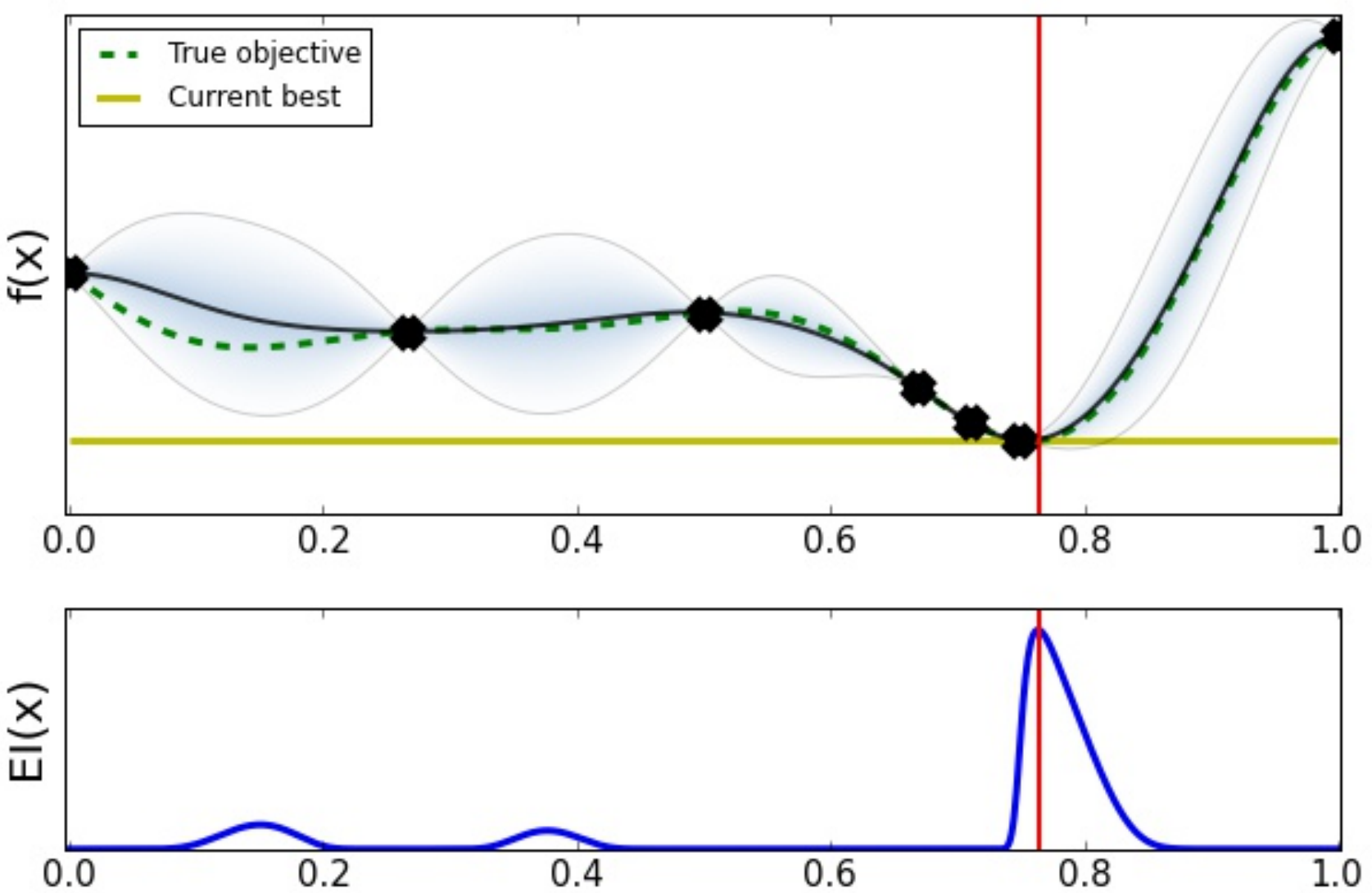
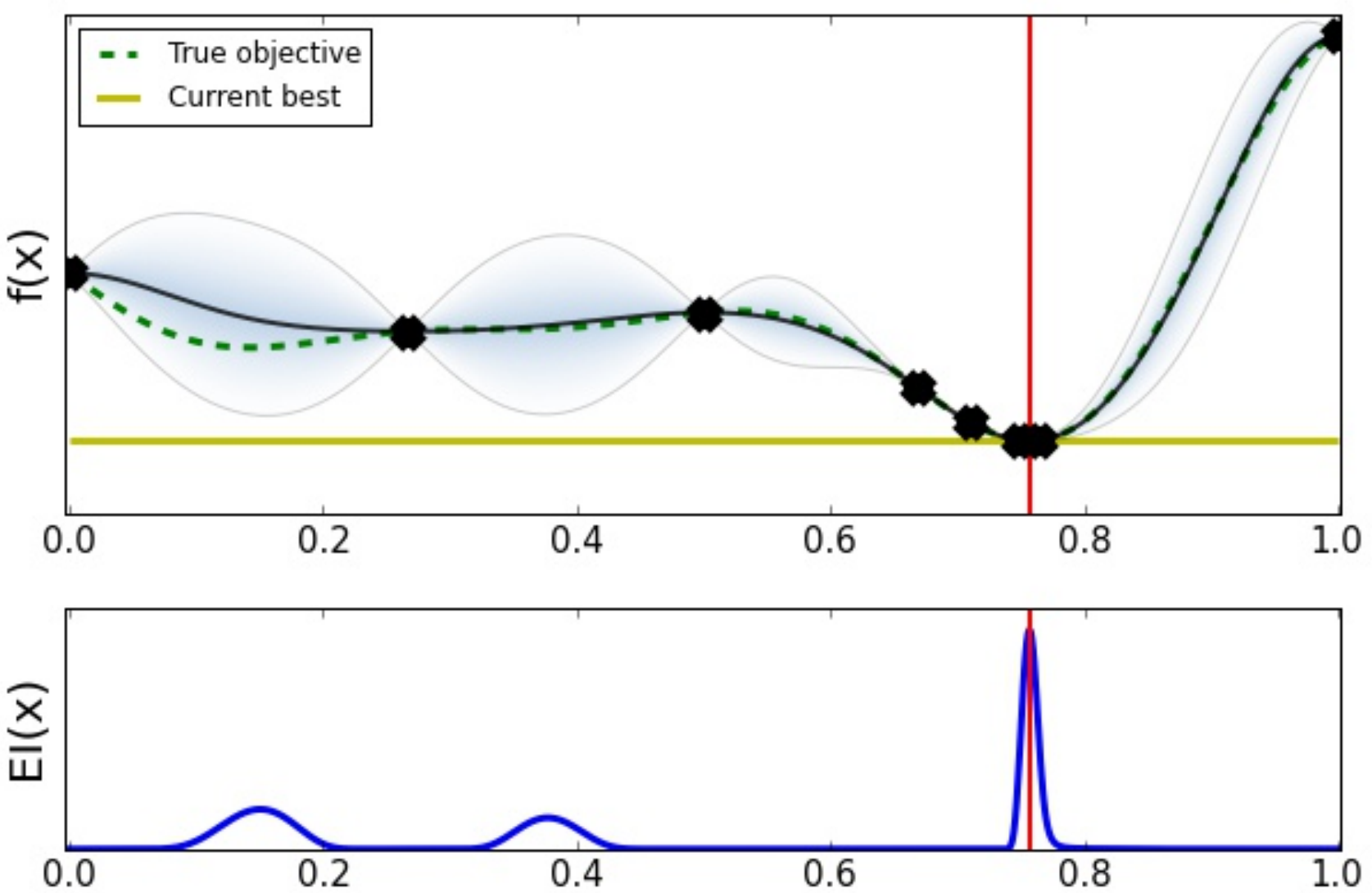


Illustration of BO



Bayesian Optimization

As a 'mapping' between two problems

BO is an strategy to transform the problem

$$x_M = \arg\min_{x \in \mathcal{X}} f(x)$$

solvable!

into a series of problems:

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha(x; \mathcal{D}_n, \mathcal{M}_n)$$

solvable!

where now:

- $\alpha(x)$ is inexpensive to evaluate.
- The gradients of $\alpha(x)$ are typically available.
- Still need to find x_{n+1}.

Recap

- Bayesian optimization is a way of encoding our beliefs about a property of a function (the minimum)
- Two key elements: the model and the acquisition function.
- Many choices in both cases, especially in terms of the acquisition function used.
- The key is to find a good balance between exploration and exploitation.

Main issues

- What to do with the hyper-parameters of the model?
- How to select points to initialize the model?
- How to optimize the acquisition function?

Questions?