Global optimisation with Gaussian processes

Javier González

Sheffield, GPSS 2016

Goal of the talk

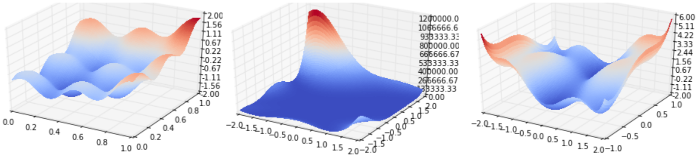
"Civilization advances by extending the number of important operations which we can perform without thinking of them."
(Alfred North Whitehead)

- To make Machine Learning completely automatic.
- To automatically design sequential experiments to optimize physical processes.

Global optimization

Consider a *well behaved* function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^D$ is (in principle) a bounded set.

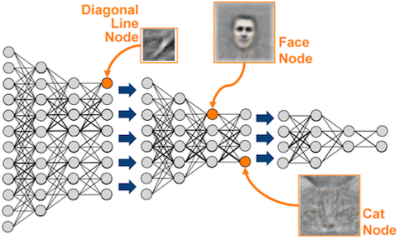
$$x_M = \arg\min_{x \in \mathcal{X}} f(x).$$



- f is explicitly unknown (computer model, process embodied in a physical process) and multimodal.
- Evaluations of f may be perturbed.
- Evaluations of f are (very) expensive.

Expensive functions, who doesn't have one?

Parameter tuning in ML algorithms.

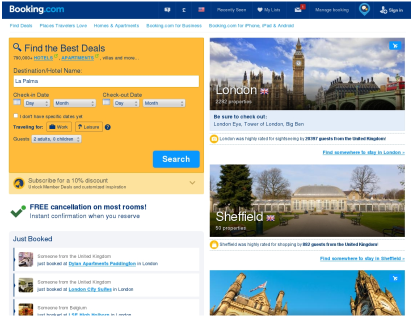


- Number of layers/units per layer
- Weight penalties
- Learning rates, etc.

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

Expensive functions, who doesn't have one?

Tuning websites with A/B testing

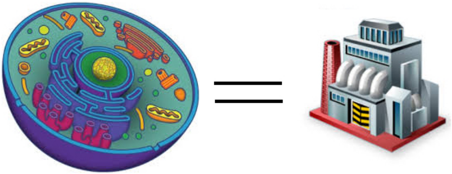


Optimize the web design to maximize sign-ups, downloads, purchases, etc.

Expensive functions, who doesn't have one?

[González, Lonworth, James and Lawrence, NIPS workshops 2014, 2015]

Design of experiments: gene optimization



- Use mammalian cells to make protein products.
- Control the ability of the cell-factory to use synthetic DNA.

Optimize genes (ATTGGTUGA...) to best enable the cell-factory to operate most efficiently.

What to do?

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

$$f(\mathbf{x}_M) - f(\mathbf{x}_{M,n}) \leq \epsilon$$

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!

Regret minimization

The goal is to make a series of $x_1, \ldots, x_N$ evaluations of f such that the *cumulative regret*

$$r_N = \sum_{n=1}^{N} f(x_{M,n}) - N f(x_M)$$

is minimized.

r_N is minimized if we start evaluating f at x_M as soon as possible.

Approach

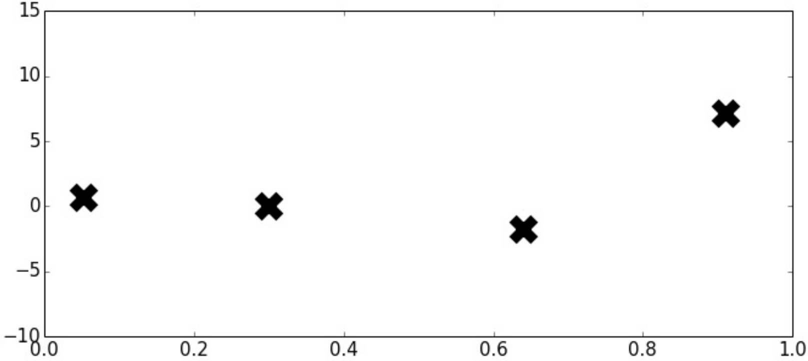
1. Minimize the regret implies to see an *optimization* problem as a *decision* problem.

2. *Decision* problems can be seen as *inference* if we take into account the *epistemic* uncertainty we have about the system we are studying.

Probability theory is the right way to model uncertainty.

Typical situation

We have a few function evaluations

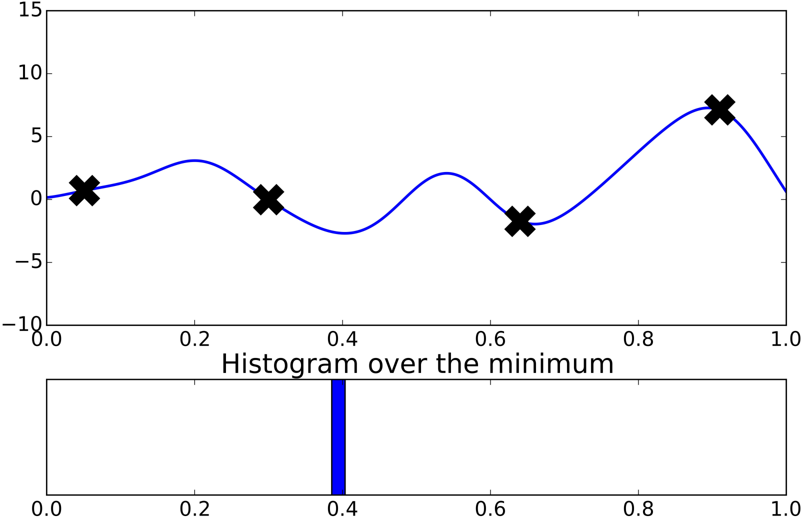


Where is the minimum of f?

Where should the take the next evaluation?

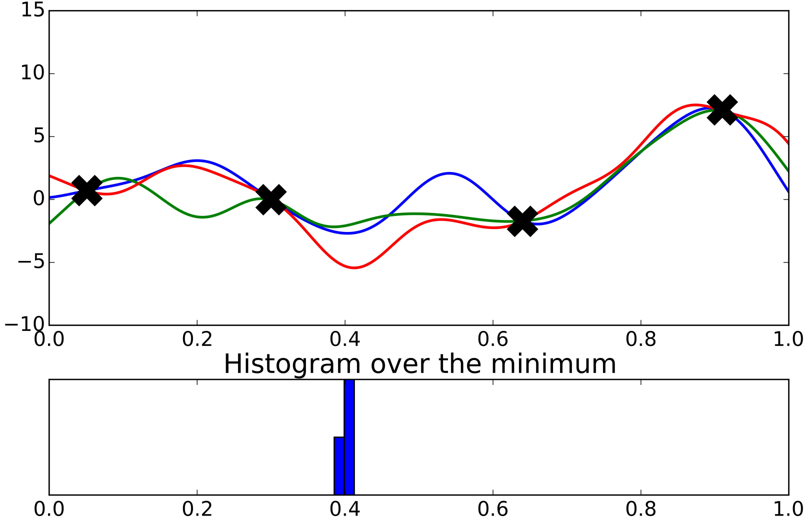
Intuitive solution

One curve



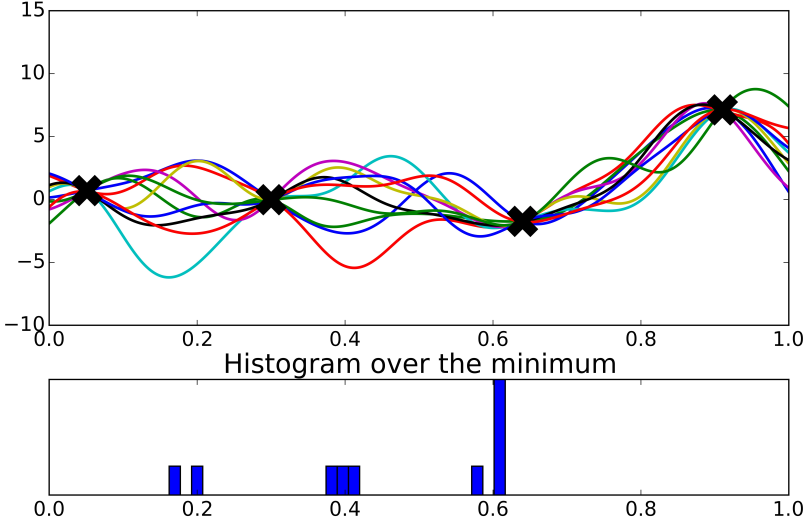
Intuitive solution

Three curves



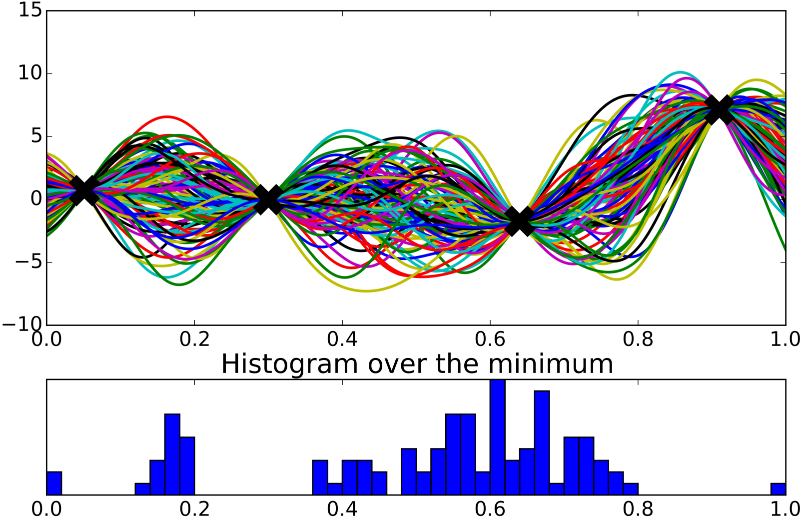
Intuitive solution

Ten curves



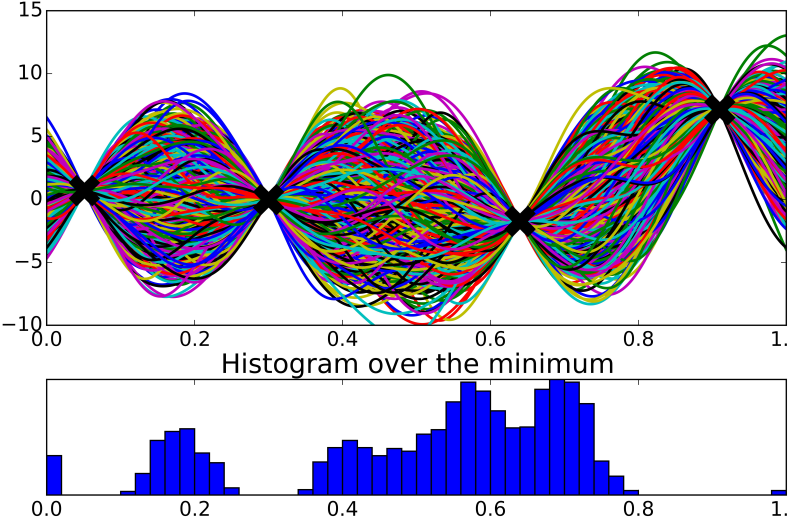
Intuitive solution

Hundred curves



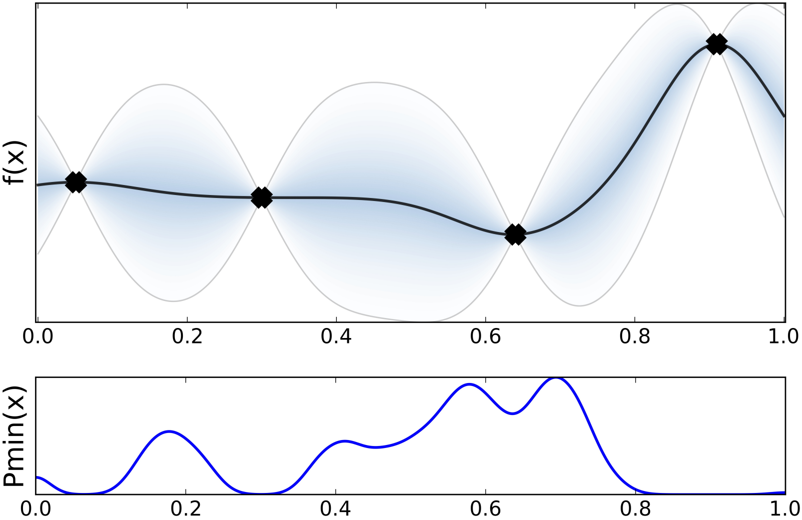
Intuitive solution

Many curves



Intuitive solution

Infinite curves



What just happened?

- We made some *prior assumptions* about our function.
- Information about the minimum is now encoded in a new function: *the probability distribution* $p_{\min}$.
- We can use $p_{\min}$ (or a functional of it) to *decide where to sample* next.
- Other functions to encode relevant information about the minimum are possible, e. g. the 'marginal expected gain' at each location.

Bayesian Optimization

Methodology to perform global optimization of multimodal black-box functions [Mockus, 1978].

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

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

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

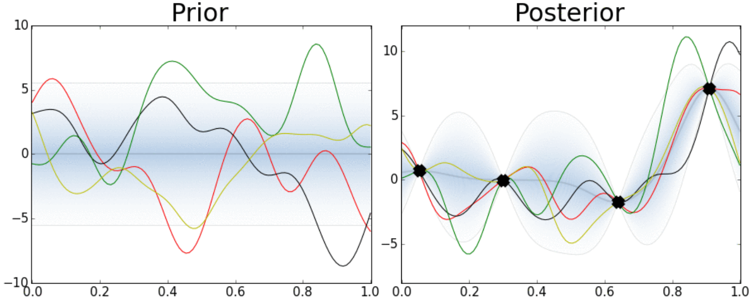
4. Augment the data.

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

Probability measure over functions

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 $\mathcal{D}$.

Acquisition functions

Making use of the model uncertainty

Here we will use Gaussian processes. GPs has marginal closed-form for the posterior mean $\mu(x)$ and variance $\sigma^2(x)$.

- **Exploration**: Evaluate in places where the variance is large.
- **Exploitation**: Evaluate in places where the mean is low.

Acquisition functions balance these two factors to determine where to evaluate next.

Exploration vs. exploitation

[Borji and Itti, 2013]

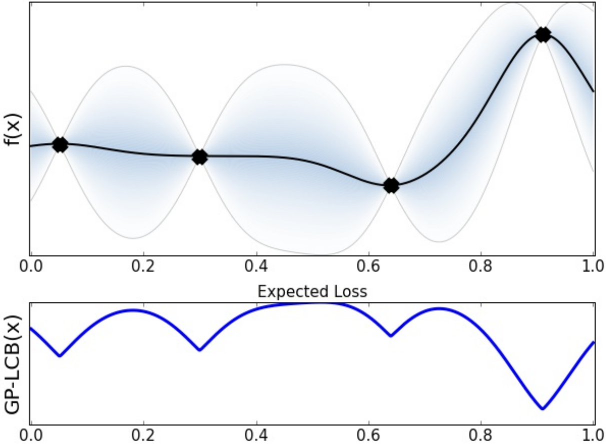
Bayesian optimization explains human active search

GP Upper (lower) Confidence Band

[Srinivas et al., 2010]

Direct balance between exploration and exploitation:

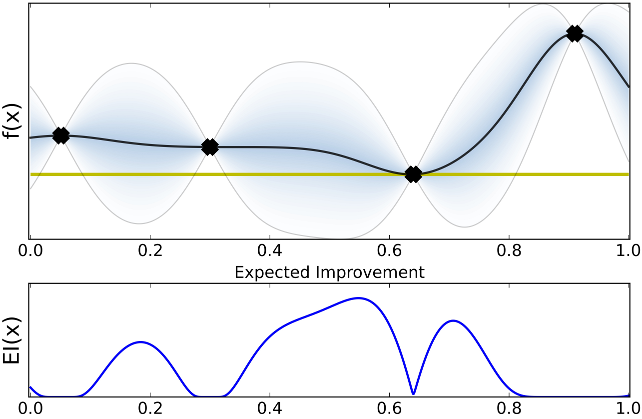
$$\alpha_{LCB}(\mathbf{x};\theta,\mathcal{D}) = -\mu(\mathbf{x};\theta,\mathcal{D}) + \beta_t\sigma(\mathbf{x};\theta,\mathcal{D})$$



Expected Improvement

[Jones et al., 1998]

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



Information-theoretic approaches

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

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

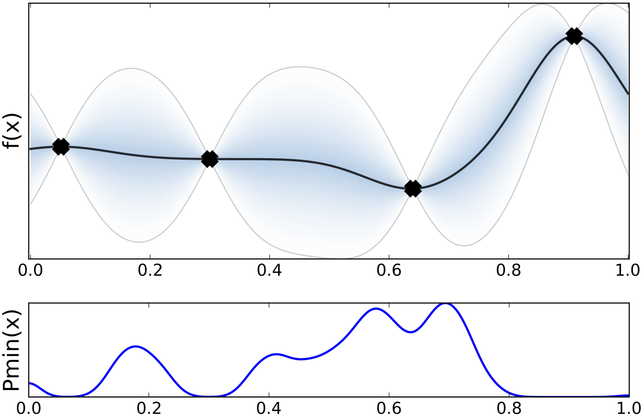


Illustration of BO

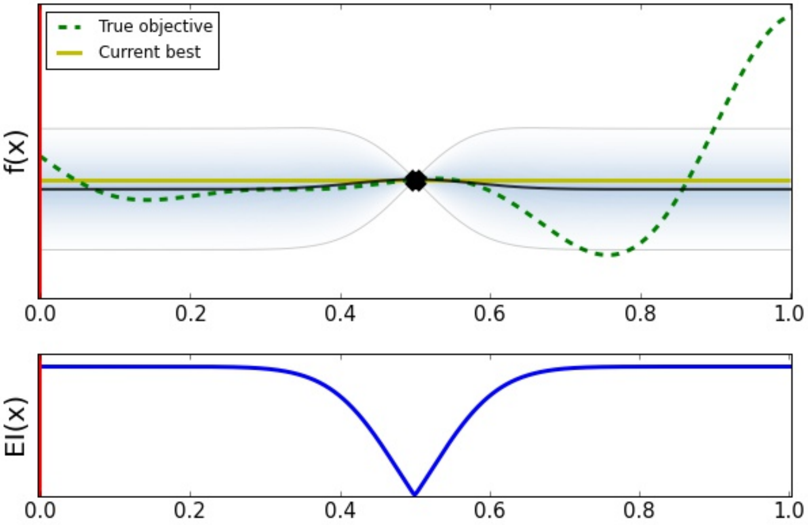


Illustration of BO

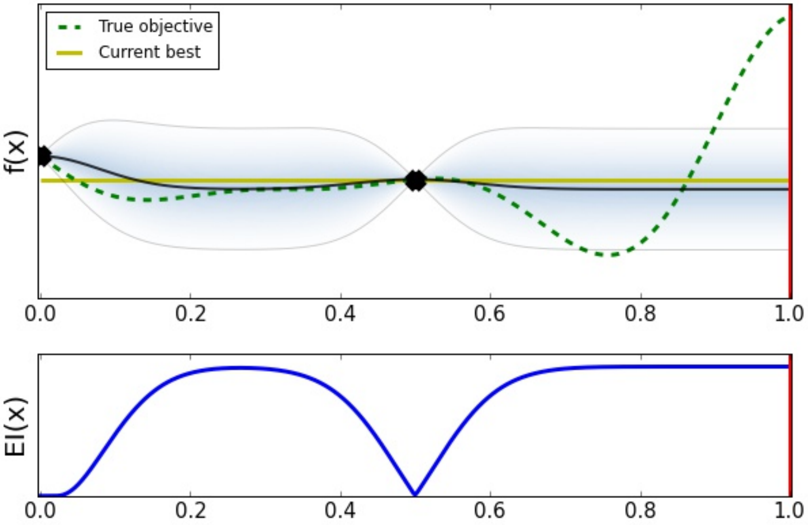


Illustration of BO

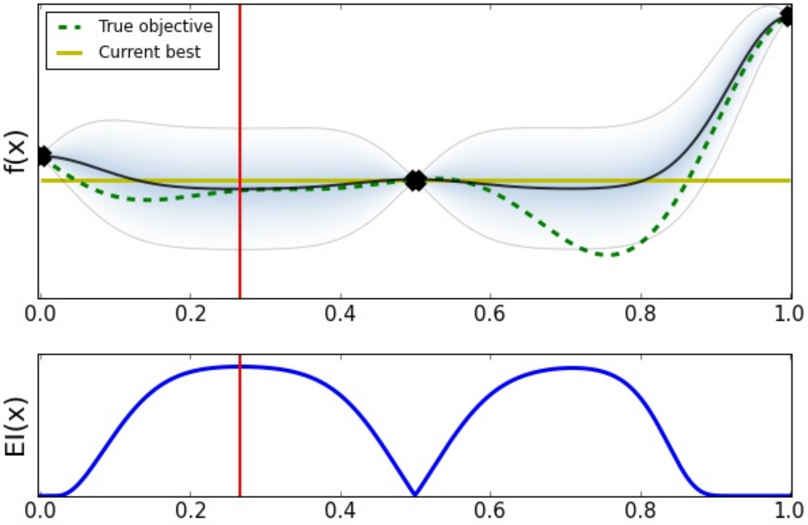


Illustration of BO

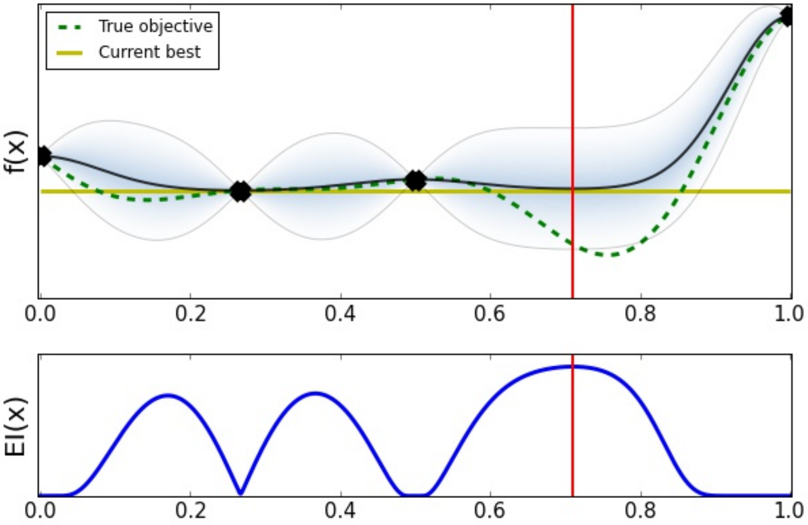


Illustration of BO

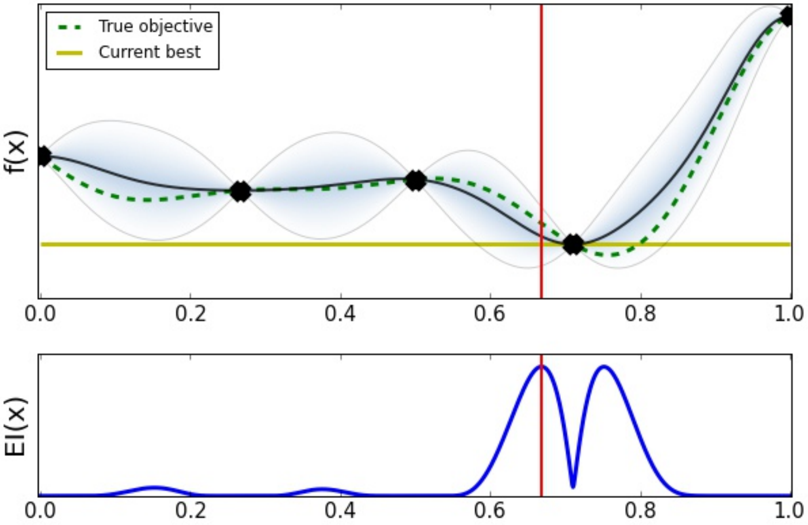


Illustration of BO

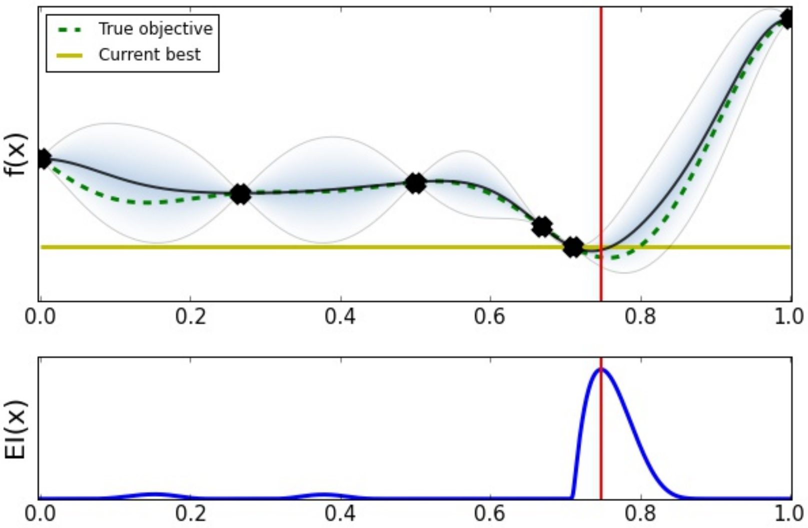


Illustration of BO

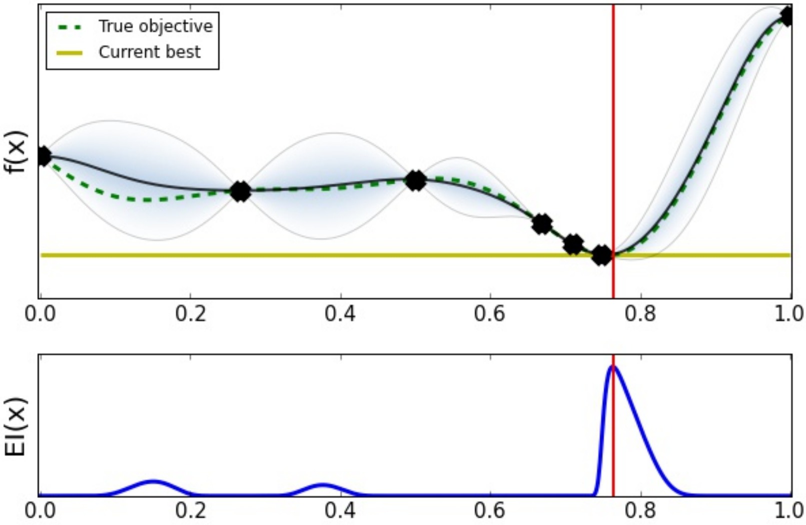


Illustration of BO

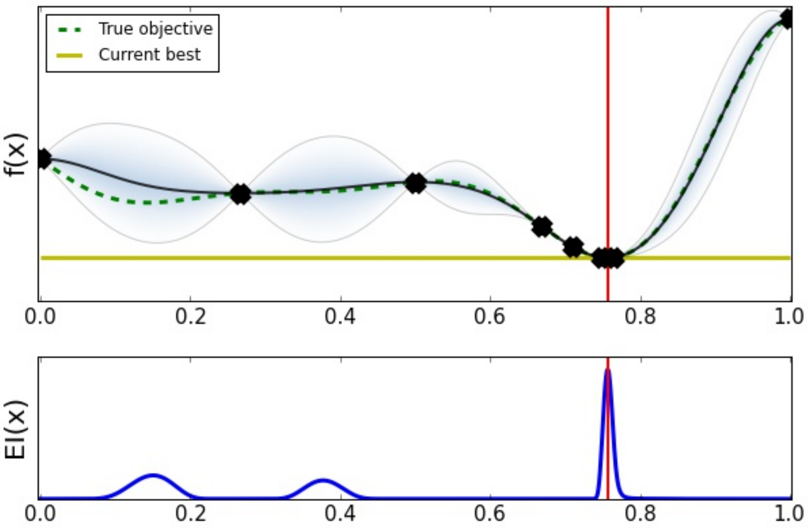
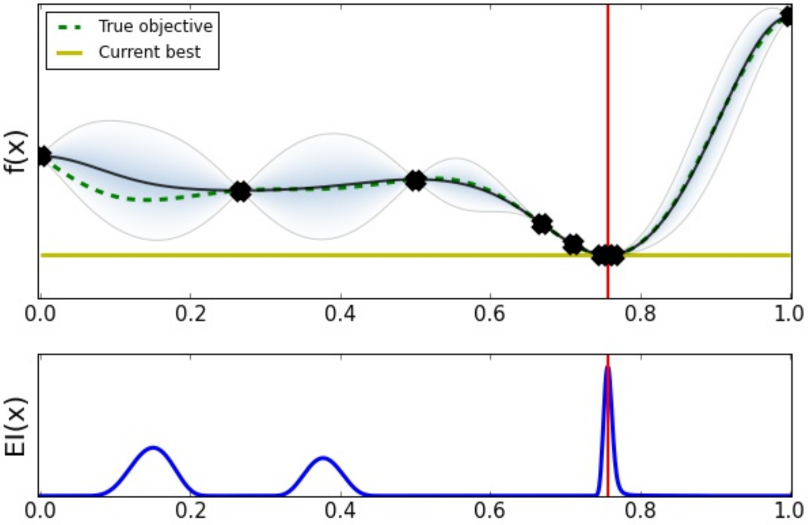


Illustration of BO



Bayesian Optimization

As a 'mapping' between two problems

BO is an strategy to transform the problem

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

unsolvable!

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}: gradient descent, DIRECT or other heuristics.

Some recent results in BO

- Parallelization
- Non-myopic methods.

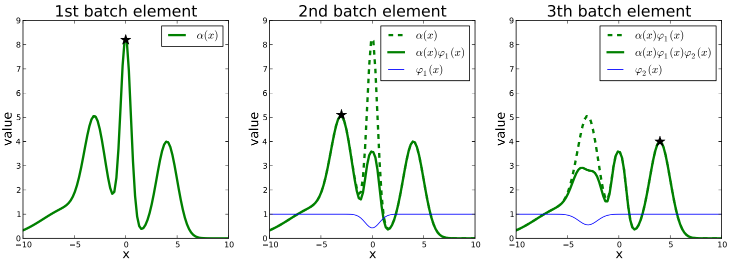
Scalable BO: Parallel/batch BO

Avoiding the bottleneck of evaluating f

- Cost of $f(\mathbf{x}_n)$ = cost of $\{f(\mathbf{x}_{n,1}), \ldots, f(\mathbf{x}_{n,nb})\}$.
- Many cores available, simultaneous lab experiments, etc.

Local penalization strategy

[González, Dai, Hennig, Lawrence, 2016]

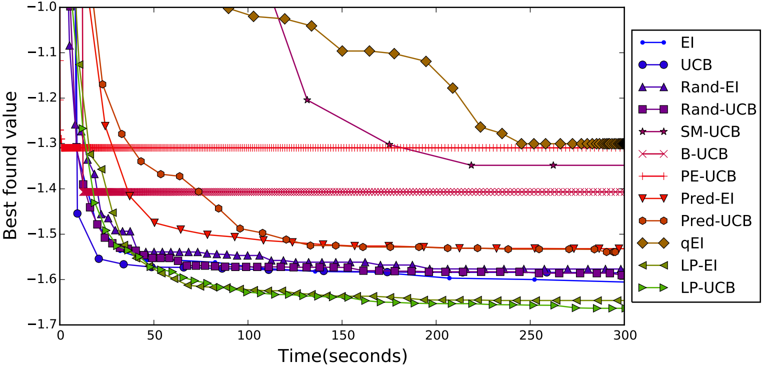


$$\mathbf{x}_{t,k} = \arg\max_{x \in \mathcal{X}} \left\{ g(\alpha(\mathbf{x}; \mathcal{I}_{t,0})) \prod_{j=1}^{k-1} \varphi(\mathbf{x}; \mathbf{x}_{t,j}) \right\},$$

g is a transformation of $\alpha(\mathbf{x}; \mathcal{I}_{t,0})$ to make it always positive.

2D experiment with 'large domain'

Comparison in terms of the wall clock time



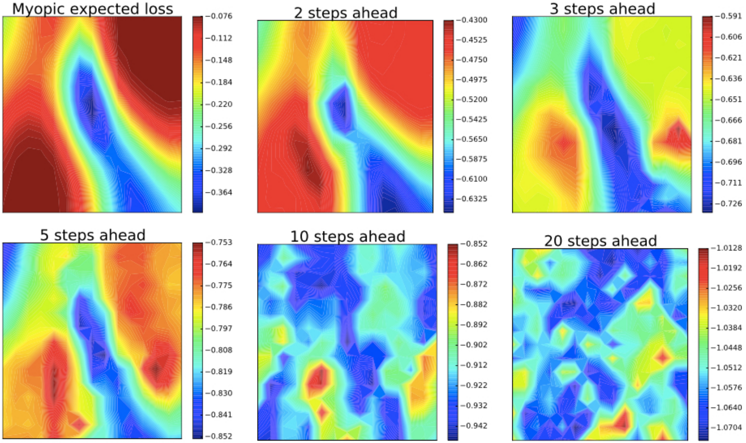
Non myopic Bayesian optimization

- Most global optimisation techniques are myopic, in considering no more than a single step into the future.
- Relieving this myopia requires solving the *multi-step lookahead* problem.

Figure: Two evaluations, if the first evaluation is made myopically, the second must be sub-optimal.

GLASSES

Global optimisation with Look-Ahead through Stochastic Simulation and Expected-loss Search [González, Osborne, Lawrence, 2016]



Automatic balance between exploration and exploitation

Results in a benchmark of objectives

	MPI	GP-LCB	EL	EL-2	EL-3	EL-5	EL-10	GLASSES
SinCos	0.7147	0.6058	0.7645	*0.8656*	0.6027	0.4881	*0.8274*	***0.9000***
Cosines	0.8637	0.8704	0.8161	*0.8423*	*0.8118*	0.7946	0.7477	***0.8722***
Branin	0.9854	0.9616	**0.9900**	0.9856	0.9673	0.9824	0.9887	0.9811
Sixhumpcamel	0.8983	**0.9346**	0.9299	0.9115	0.9067	0.8970	0.9123	0.8880
Mccormick	**0.9514**	0.9326	0.9055	*0.9139*	*0.9189*	*0.9283*	*0.9389*	*0.9424*
Dropwave	0.7308	0.7413	0.7667	0.7237	0.7555	0.7293	0.6860	***0.7740***
Powers	0.2177	0.2167	0.2216	*0.2428*	*0.2372*	*0.2390*	*0.2339*	***0.3670***
Ackley-2	0.8230	**0.8975**	0.7333	0.6382	0.5864	0.6864	0.6293	0.7001
Ackley-5	0.1832	0.2082	0.5473	*0.6694*	0.3582	0.3744	***0.6700***	0.4348
Ackley-10	0.9893	0.9864	0.8178	*0.9900*	*0.9912*	***0.9916***	*0.8340*	*0.8567*
Alpine2-2	**0.8628**	0.8482	0.7902	0.7467	0.5988	0.6699	0.6393	0.7807
Alpine2-5	0.5221	0.6151	**0.7797**	0.6740	0.6431	0.6592	0.6747	0.7123

Wrapping up

- BO is fantastic tool for global parameter optimization in ML and experimental design.
- The model and the acquisition function are the two most important bits.
- Non myopic approach are needed to find good balance between exploration and exploitation.
- Software available! Use GPyOpt!