Course in Bayesian Optimization

Javier González

University of Sheffield, Sheffield, UK

28th October 2015

Recap

Yesterday we discussed:

- ML as optimization and ML as probabilistic modeling.
- In the second case we also need to 'optimize parameters'
- Probability theory provides a mathematical framework to deal with uncertainty.
- Gaussian processes are a fundamental way to model uncertainty.

Parameter optimization is crucial in any framework

Today’s agenda

- In what cases we can make an explicit use of the epistemic uncertainty to make decisions?
- Global optimization, different strategies.
- Probabilistic models to solve global optimization problems.

Parameter optimization is crucial in any ML framework

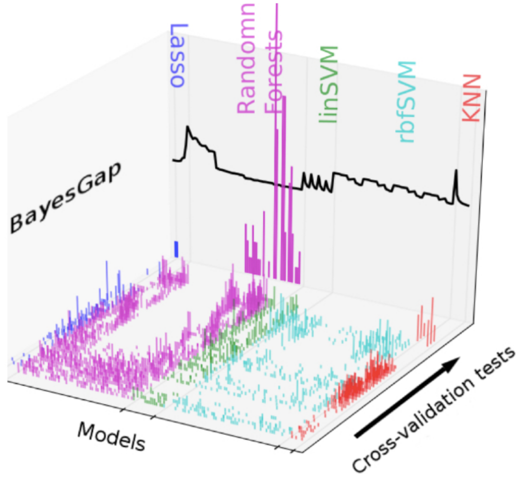
Goal of the day

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

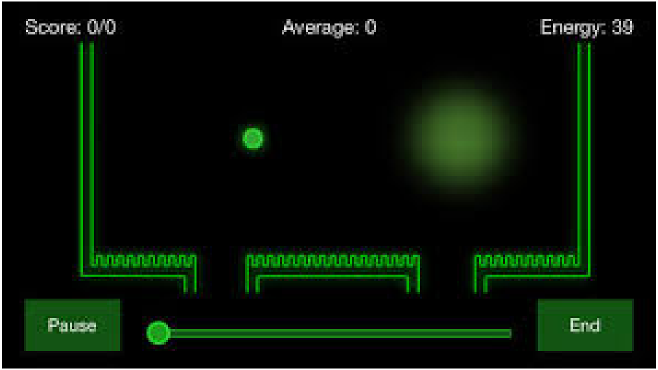
To make ML completely automatic.

Goal of the day

[Hoffman, Shahriari and de Freitas, 2013]



Kappenball: using the uncertainty to make optimal decisions.

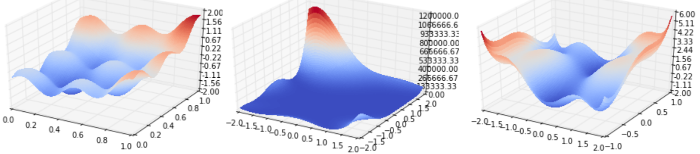


Making optimal choices using epistemic uncertainty.

Global optimization

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

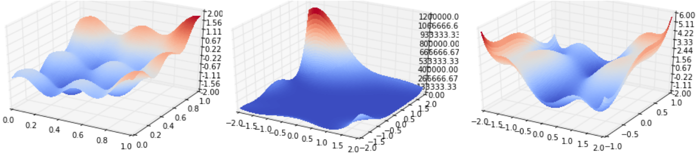
$$x_M = \arg\min_{x \in \mathcal{X}} f(x).$$



Global optimization

Consider a ‘well behaved’ function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^D$ is a compact set.

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



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

What to do?

Option 1: Previous knowledge

To use what we know about the problem. 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 $\mathbf{x}_{M,n}$ such that

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

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

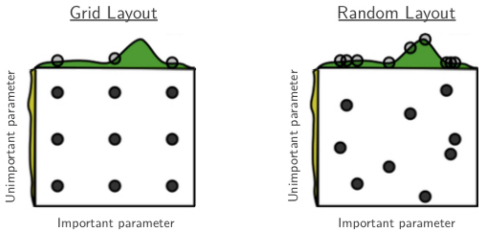
$$(L/\epsilon)^D evaluations!$$

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

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?

Regret minimization

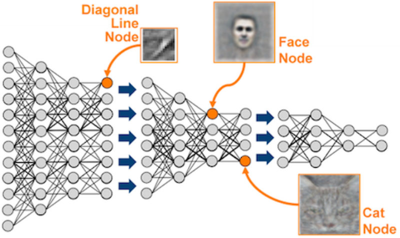
The goal is to make a series of $x_1, \ldots, x_N$ evaluations of f such that the *cumulative regret*

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

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

Expensive functions, who doesn't have one?

Parameter tuning in ML algorithms.

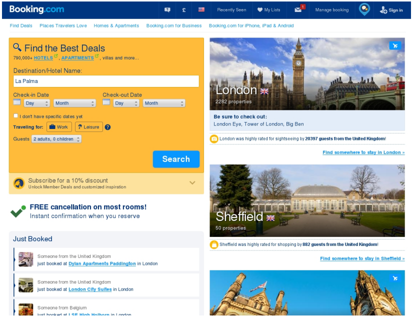


- Number of layers/units per layer
- Weight penalties
- Learning rates, etc.

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

Expensive functions, who doesn't have one?

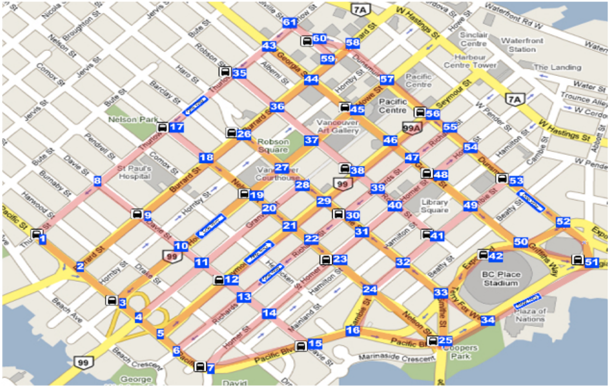
Tuning websites with A/B testing



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

Expensive functions, who doesn’t have one?

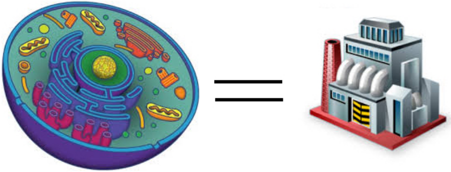
Active Path Finding in Middle Level



Optimise the location of a sequence of waypoints in a map to navigate from a location to a destination.

Expensive functions, who doesn't have one?

Synthetic gene design: Use mammalian cells to make protein products.



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

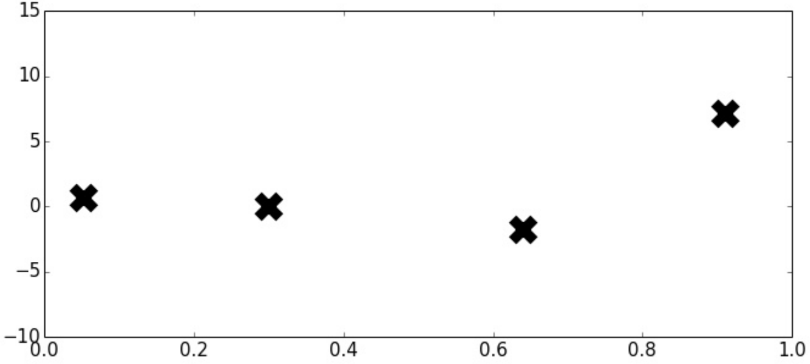
Expensive functions, who doesn't have one?

Many other problems:

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

Typical situation

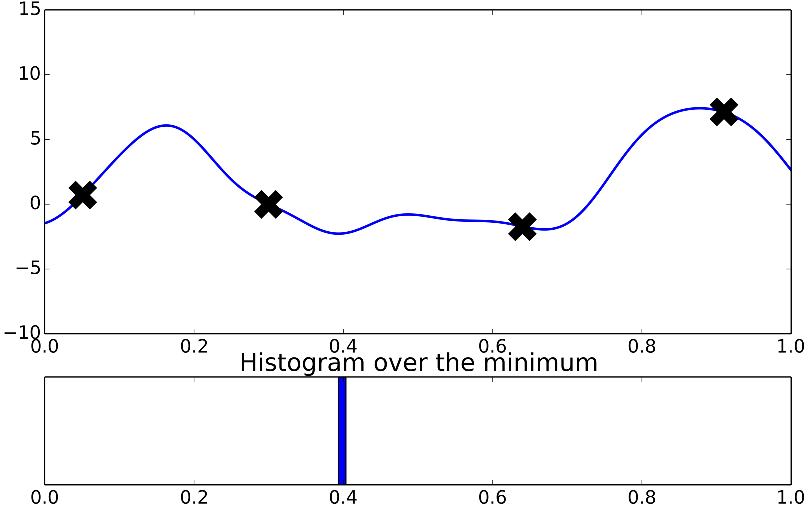
We have a few function evaluations



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

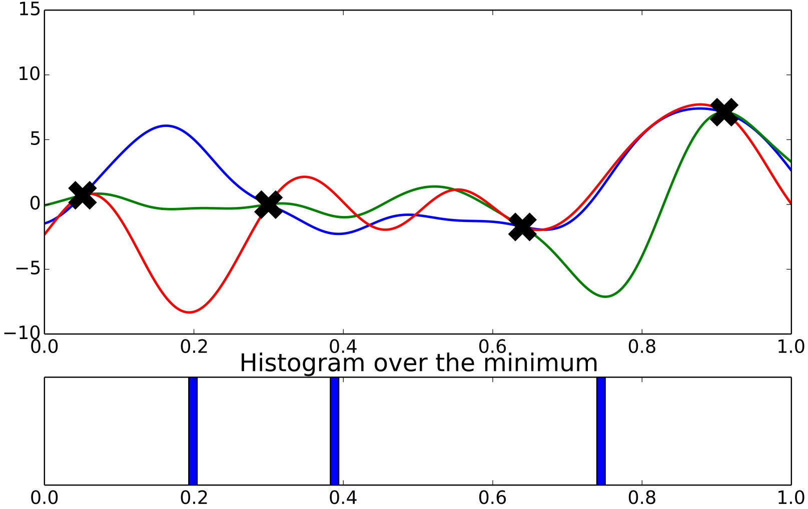
Intuitive solution

One curve



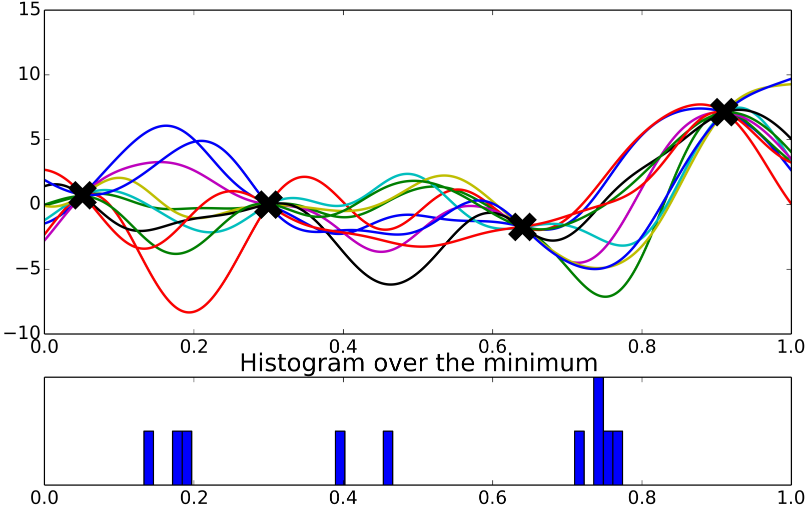
Intuitive solution

Three curves



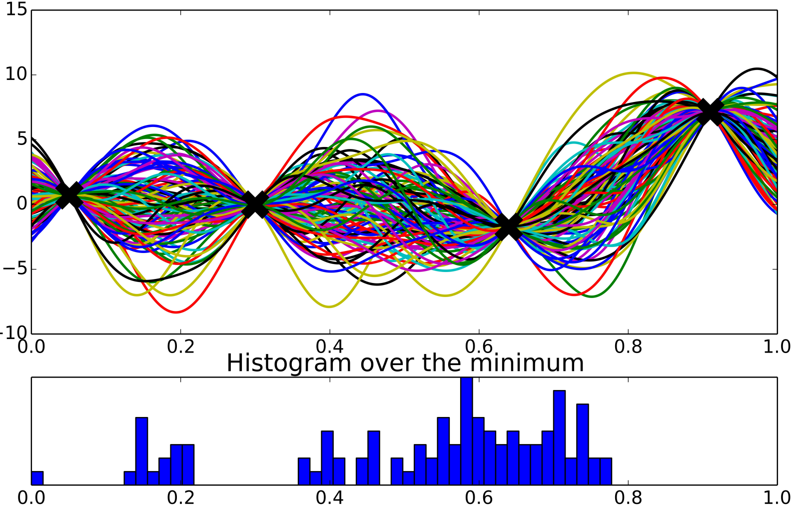
Intuitive solution

Ten curves



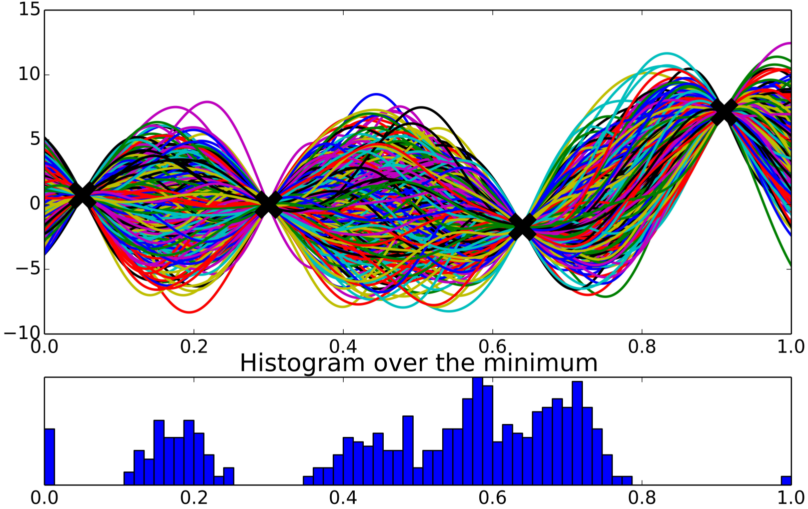
Intuitive solution

Hundred curves



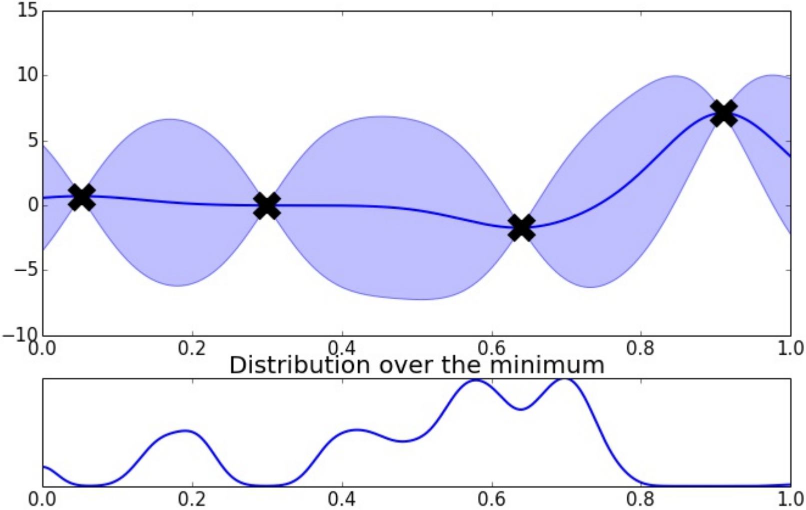
Intuitive solution

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 $p_{\min}$ in this case).
- We can use $p_{\min}$ (or a functional of it: entropy search) 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*.

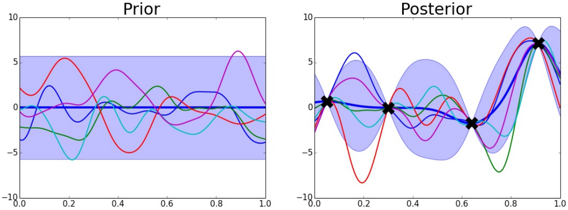
4. Augment the data.

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

Probability measure over functions

Default Choice: Gaussian processes

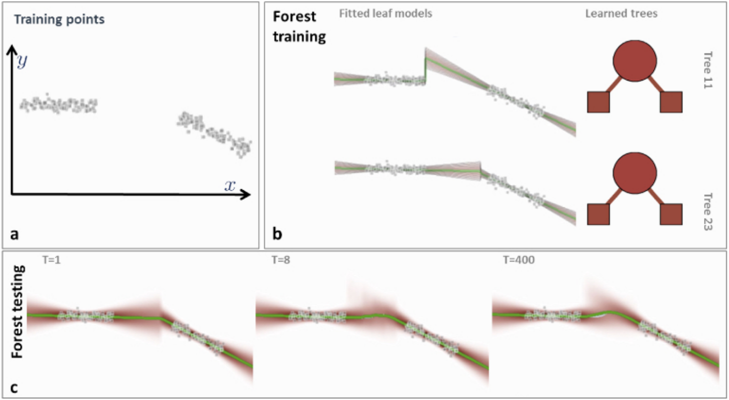
Infinite-dimensional probability density, such that each linear finite-dimensional restriction is multivariate Gaussian.



- 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 $\mathcal{D}$.

Other models are also possible: Random Forrest

[Criminisi et al, 2011]



Other models are also possible: t-Student processes

Student-t Processes as Alternatives to Gaussian Processes

Amar Shah
University of Cambridge

Andrew Gordon Wilson
University of Cambridge

Zoubin Ghahramani
University of Cambridge

Abstract

We investigate the Student-t process as an alternative to the Gaussian process as a non-parametric prior over functions. We derive closed form expressions for the marginal likelihood and predictive distribution of a Student-t process, by integrating away an

simple exact learning and inference procedures, and impressive empirical performances [Rasmussen, 1996], Gaussian processes as kernel machines have steadily grown in popularity over the last decade.

At the heart of every Gaussian process (GP) is a parametrized covariance kernel, which determines the properties of likely functions under a GP. Typically simple parametric kernels, such as the Gaus-

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

Bayesian optimization explains human active search

[Borji and Itti, 2013]

Exploration vs. exploitation

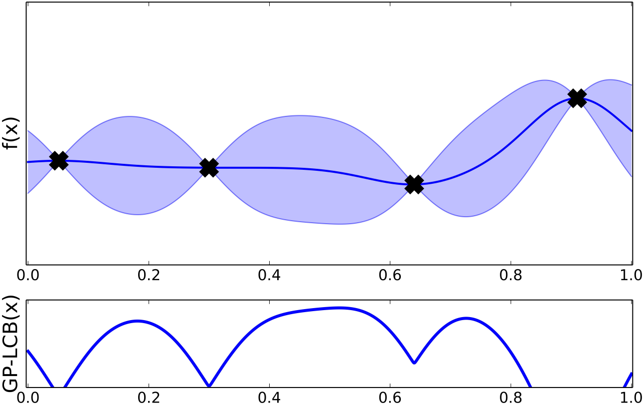
Picture source: http://peakdistrictcycleways.co.uk

GP Upper (lower) Confidence Band

[Srinivas et al., 2010]

Direct balance between exploration and exploitation:

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



GP Upper (lower) Confidence Band

[Srinivas et al., 2010]

- In noiseless cases, it is a lower bound of the function to minimize.
- This allows to computer a bound on how close we are to the minimum.
- Optimal choices available for the ’regularization parameter’.

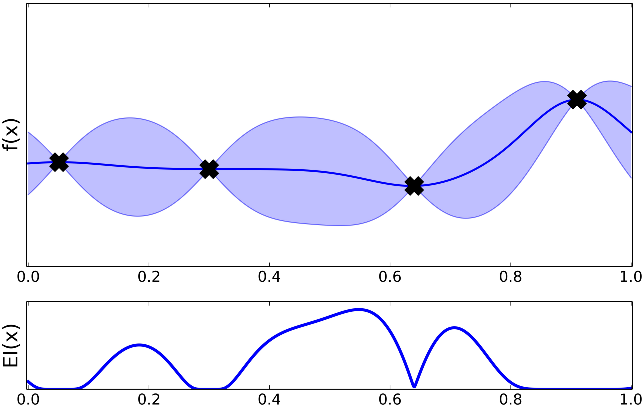
Theorem 1 *Let* $\delta \in (0,1)$ *and* $\beta_t = 2\log(|D|t^2\pi^2/6\delta)$. *Running GP-UCB with* β_t *for a sample* f *of a GP with mean function zero and covariance function* $k(\boldsymbol{x}, \boldsymbol{x}')$, *we obtain a regret bound of* $\mathcal{O}^*(\sqrt{T\gamma_T \log|D|})$ *with high probability. Precisely, with* $C_1 = 8/\log(1+\sigma^{-2})$ *we have*

$$\Pr\left\{R_T \leq \sqrt{C_1 T \beta_T \gamma_T} \quad \forall T \geq 1\right\} \geq 1 - \delta.$$

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

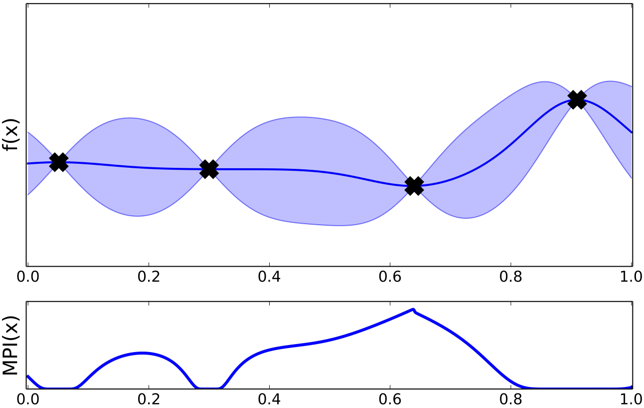
$$\gamma(x) = \frac{f(x_{best}) - \mu(\mathbf{x};\theta,\mathcal{D}) + \psi}{\sigma(\mathbf{x};\theta,\mathcal{D})}.$$

Maximum Probability of Improvement

[Hushner, 1964]

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

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



Maximum Probability of Improvement

[Hushner, 1964]

- First used acquisition: very intuitive.
- Less used in practice.
- Explicit for available for Gaussian posteriors.

$$\alpha_{MPI}(\mathbf{x};\theta,\mathcal{D}) = \Phi(\gamma(x))).$$

where

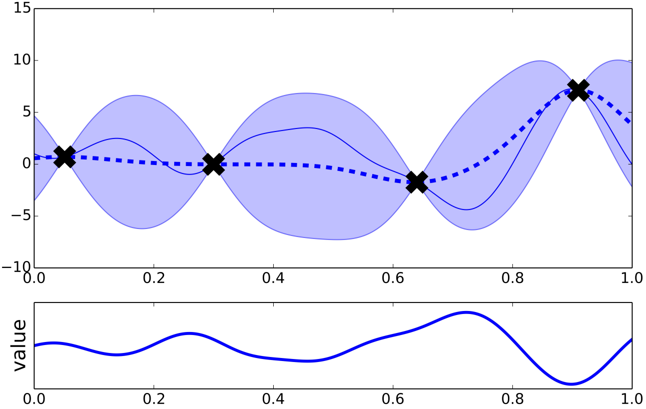
$$\gamma(x) = \frac{f(x_{best}) - \mu(\mathbf{x};\theta,\mathcal{D}) + \psi}{\sigma(\mathbf{x};\theta,\mathcal{D})}.$$

Thomson sampling

Probability matching

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

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



Thomson sampling

Probability matching [Rahimi and B. Recht, 2007]

- It is easy to generate posterior samples of a GP at a finite set of locations.
- More difficult is to generate 'continuous' samples.

Possible using the Bochner's lemma: existence of the Fourier dual of k, $s(\omega)$ which is equal to the spectral density of k

$$k(x, x') = \nu \mathbb{E}_{\omega}\left[e^{-i\omega^T(x-x')}\right] = 2\nu \mathbb{E}_{\omega,b}\left[\cos(\omega x^T + b)\cos(\omega x^T + b)\right]$$

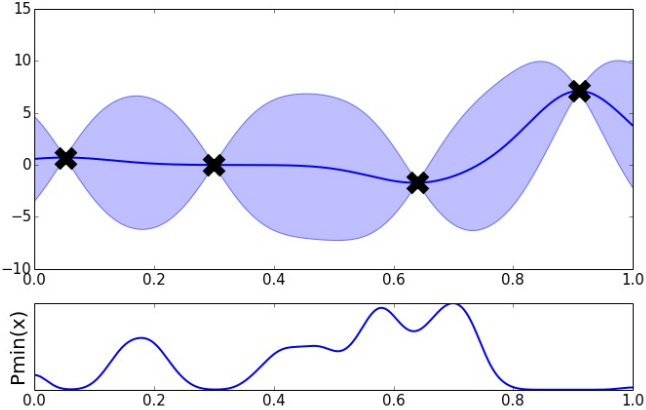
With sampling and this lemma (taking $p(w) = s(\omega)/\nu$ and $b \sim \mathcal{U}[0, 2\pi]$) we can construct a feature based approximation for sample paths of the GP.

$$k(x, x') \approx \frac{\nu}{m} \sum_{i=1}^{m} e^{-i\omega^{(i)T}x} e^{-i\omega^{(i)T}x'}$$

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

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

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 $p_{min}(x)$.

Illustration of BO

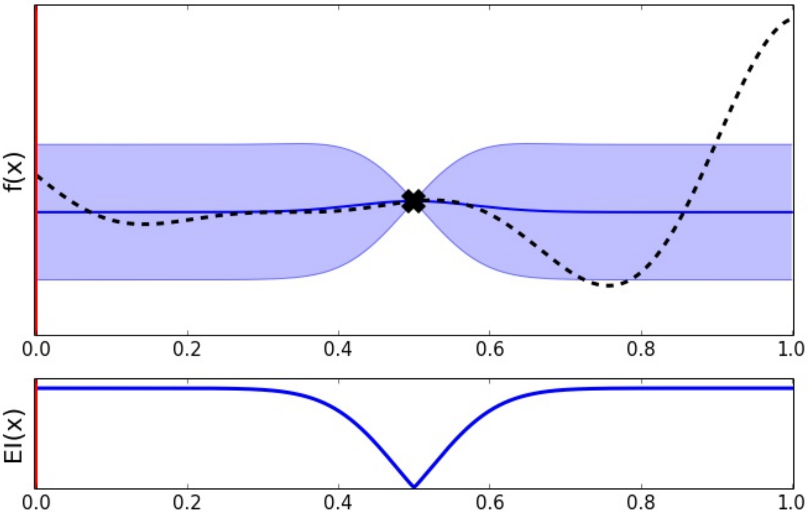


Illustration of BO

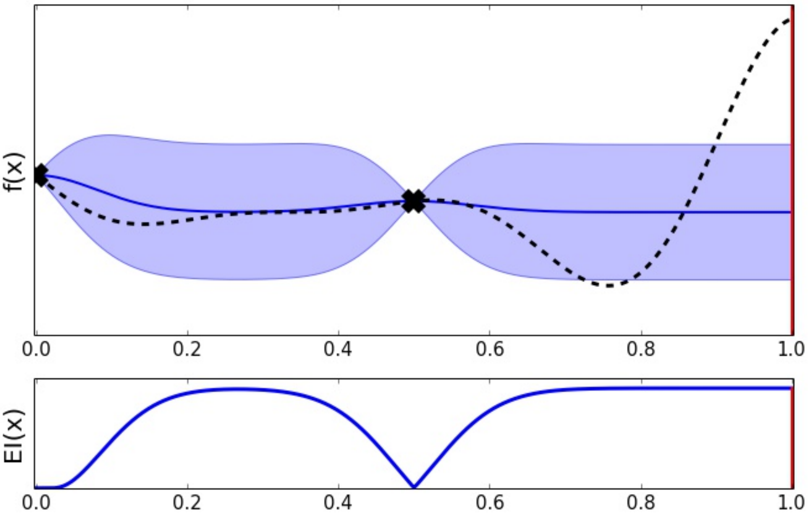


Illustration of BO

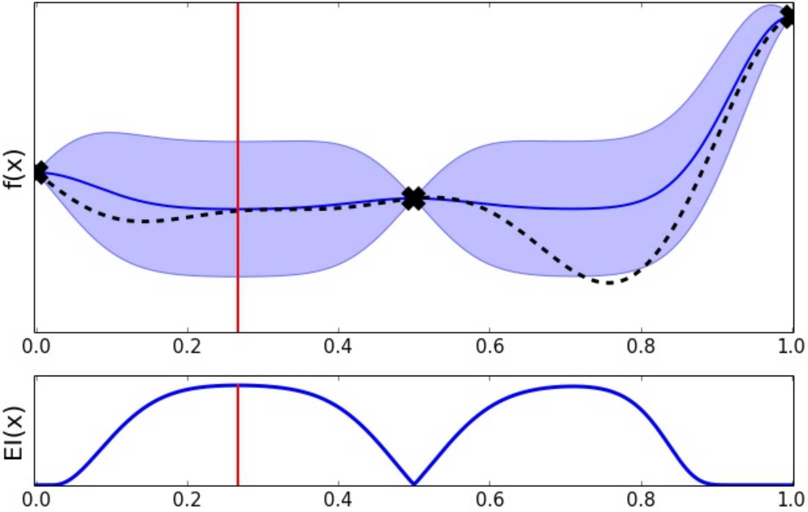


Illustration of BO

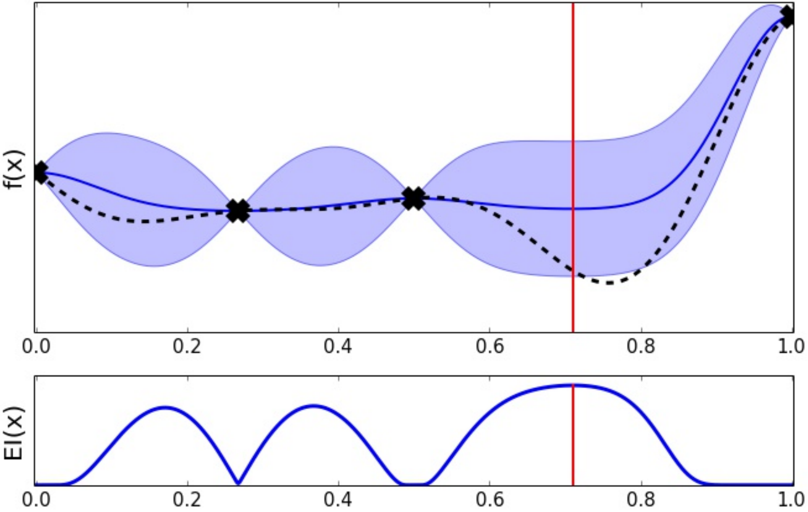


Illustration of BO

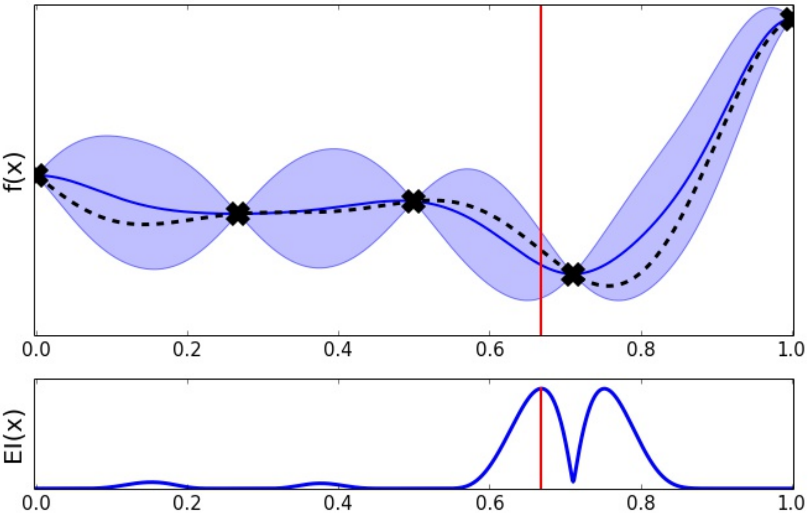


Illustration of BO

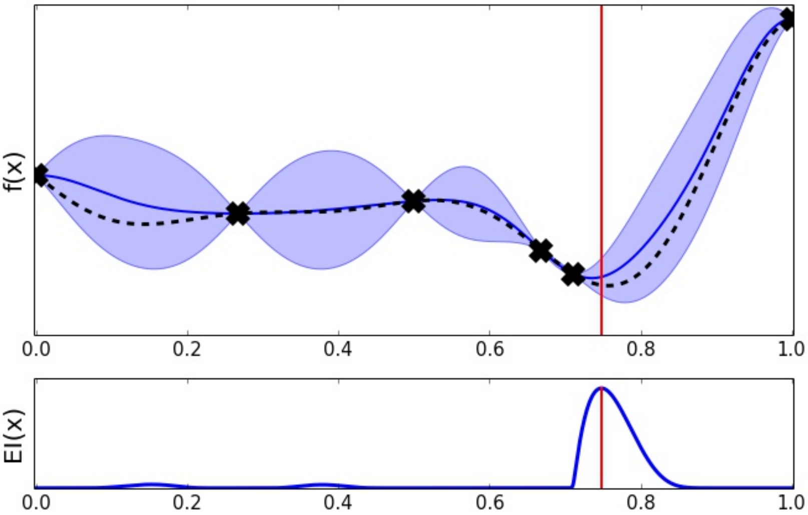


Illustration of BO

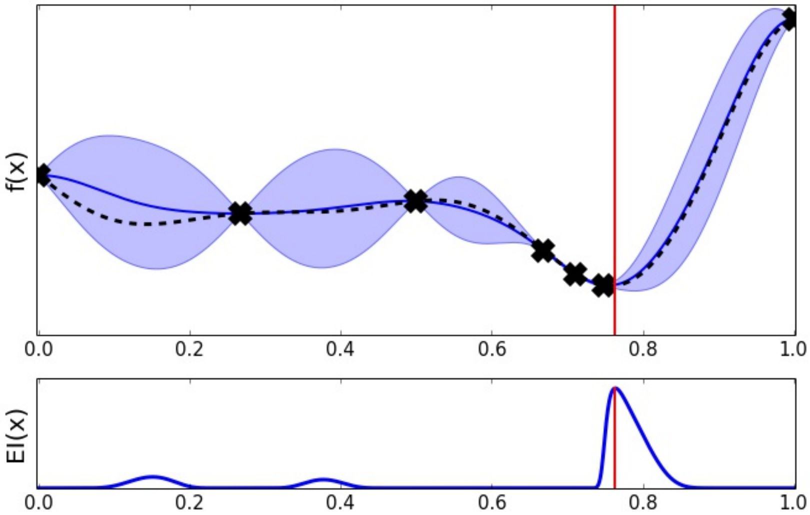


Illustration of BO

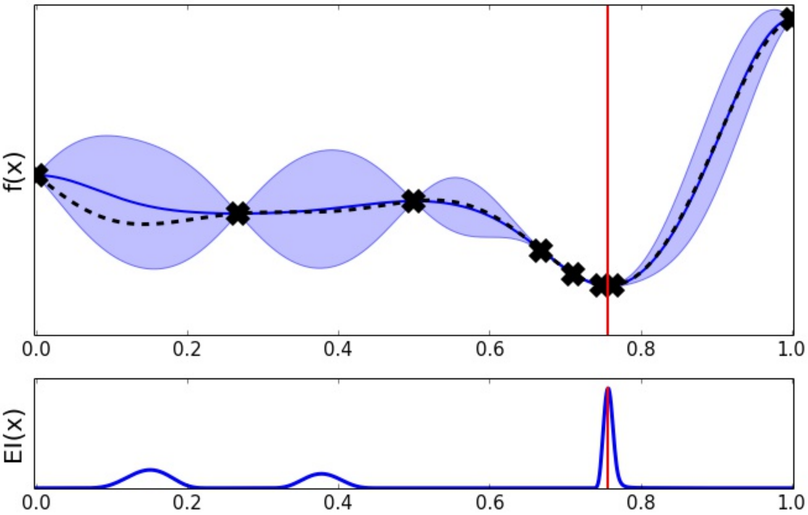
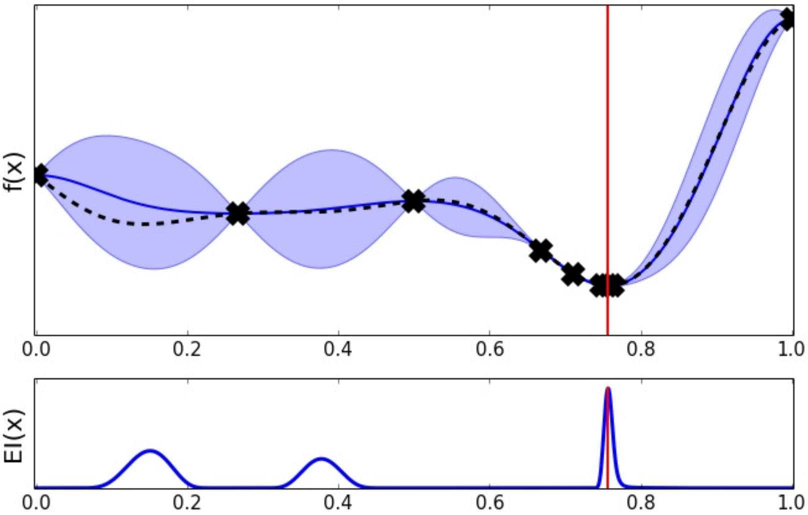


Illustration of BO



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, specially in terms of the acquisition function used.
- The key is to find a good balance between exploration and exploitation.

Multi-armed bandits

Problem in which a gambler at a row of slot machines has to decide which machines to play given that each one returns a benefit according to a probability distribution.

Multi-armed bandits

- Assume that the machines are correlated.

Multi-armed bandits

- Assume that the machines are correlated.
- Assume that the distribution over the benefits is multivariate Gaussian.

Multi-armed bandits

- Assume that the machines are correlated.
- Assume that the distribution over the benefits is multivariate Gaussian.
- Increase the number machines: take the limit case $n \to \infty$.

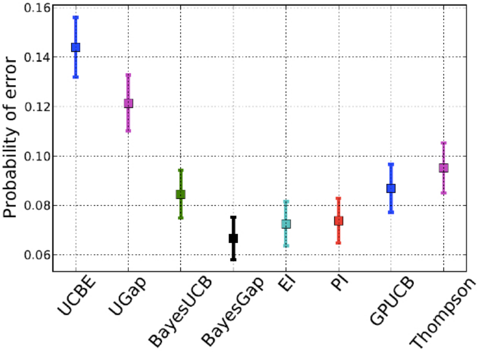
Multi-armed bandits

- Assume that the machines are correlated.
- Assume that the distribution over the benefits is multivariate Gaussian.
- Increase the number machines: take the limit case $n \to \infty$.
- Bayesian optimization!

The choice of utility in practice

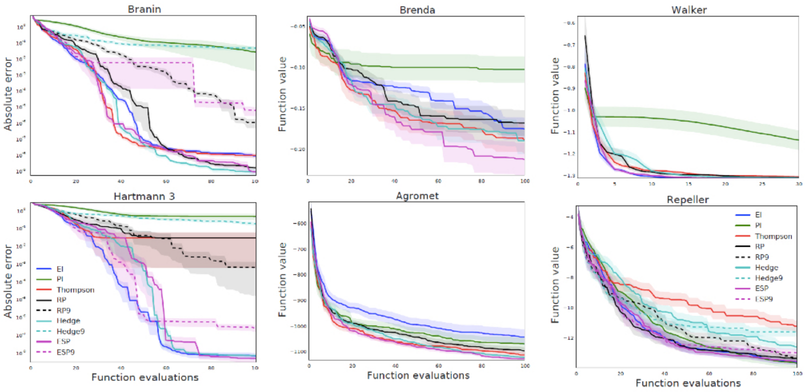
[Hoffman, Shahriari and de Freitas, 2013]

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.

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}.

Methods to optimise the acquisition function

This may not be easy.

- Gradient descent methods: Conjugate gradient, BFGS, etc.
- Liptchiz based heuristics: DIRECT.
- Evolutionary algorithms: CMA.

Some of these methods can also be used to directly optimize f

Gradient descent

[Avriel,2013], but many others

Algorithm 2: Gradient Descent

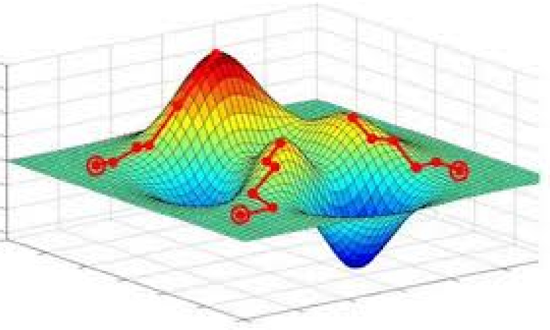
input : $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a differentiable function
$\mathbf{x}^{(0)}$ an initial solution

output: $\mathbf{x}^{\star}$, a local minimum of the cost function f.

begin
  $k \leftarrow 0$;
  while STOP-CRIT **and** $(k < k_{max})$ **do**
    $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x})$;
    with $\alpha^{(k)} = \arg \min_{\alpha \in \mathbb{R}_+} f\big(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x})\big)$;
    $k \leftarrow k + 1$;
  return $\mathbf{x}^{(k)}$
end

We need to know the gradients. This is the case for most acquisitions but not for all of them (PES for instance).

Gradient descent



May fall in local minima if the function is multimodal: multiple initialisations.

‘DIviding RECTangles’, DIRECT

[Perttunen at al. 1993]

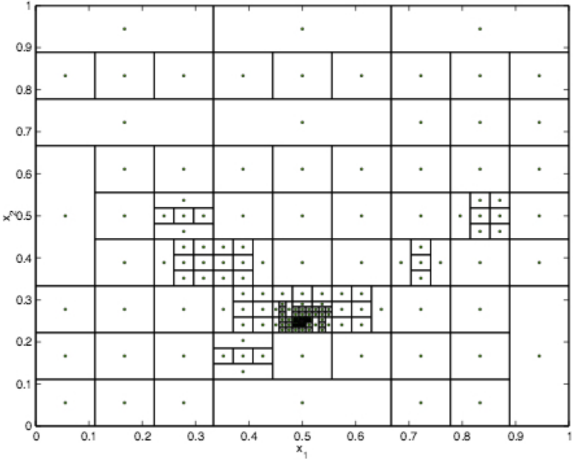
Algorithm `DIRECT`(’myfcn’,bounds,opts)

Normalize the domain to be the unit hyper-cube with center c_1
Find $f(c_1)$, $f_{min} = f(c_1)$, $i = 0$, $m = 1$
Evaluate $f(c_1 \pm \delta e_i$, $1 \leq i \leq n$, and divide hyper-cube
while $i \leq maxits$ and $m \leq maxevals$ **do**
    Identify the set S of all pot. optimal rectangles/cubes
    for all $j \in S$
        Identify the longest side(s) of rectangle j
        Evaluate myfcn at centers of new rectangles, and divide j into smaller rectangles
        Update f_{min}, $xatmin$, and m
    end for
    $i = i + 1$
end while

Minimal hypothesis about the acquisition

‘DIviding RECTangles’, DIRECT

[Perttunen at al. 1993]

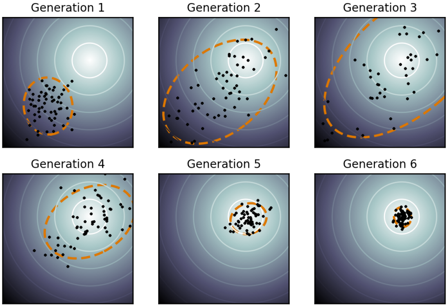


Finds good solution in general and doesn’t need gradient. Not generalizable to non-squared domains.

Covariance Matrix Adaptation, CMA

[Hansen and Ostermeier, 2001].

- Sample for a Gaussian with some mean μ and covariance matrix Σ.
- Select the best points and use them to update μ and Σ.
- Sample form the new Gaussian.



BO vs other methods

[Osborne et al, 2009]

Bayesian optimization works better in practice!

				GPGO 1-Step		GPGO 2-Step
	EGO	RBF	DIRECT	Non-Periodic	Periodic	Non-Periodic
Br	0.943	0.960	0.958	**0.980**	—	—
C6	0.962	0.962	0.940	0.890	—	**0.967**
G–P	0.783	0.815	**0.989**	0.804	—	**0.989**
H3	0.970	0.867	0.868	**0.980**	—	—
H6	0.837	0.701	0.689	**0.999**	—	—
Sh5	0.218	0.092	0.090	**0.485**	—	—
Sh7	0.159	0.102	0.099	**0.650**	—	—
Sh10	0.135	0.100	0.100	**0.591**	—	—
GK2	0.571	0.567	0.538	**0.643**	—	—
GK3	0.519	0.207	0.368	**0.532**	—	—
Shu	**0.492**	0.383	0.396	0.437	0.348	0.348
G2	0.979	**1.000**	0.981	**1.000**	**1.000**	—
G5	**1.000**	0.998	0.908	0.925	0.957	—
A2	0.347	0.703	0.675	0.606	0.612	**0.781**
A5	0.192	**0.381**	0.295	0.089	0.161	—
R	0.652	0.647	0.776	0.675	**0.933**	—
mean	0.610	0.593	0.604	**0.705**	—	—

Robotics video

How to initialise the model?

- One point in the centre of the domain.
- Uniformly selected random locations.
- Latin design.
- Halton sequences.
- Determinantal point processes.

The idea is always to start at some locations trying to minimise the initial model uncertainty.

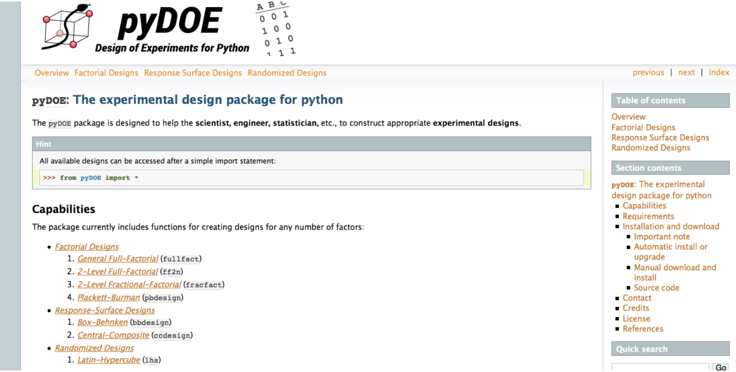
Latin design

$n \times n$ array filled with n different symbols, each occurring exactly once in each row and exactly once in each column.

A	B	F	C	E	D
B	C	A	D	F	E
C	D	B	E	A	F
D	E	C	F	B	A
E	F	D	A	C	B
F	A	E	B	D	C

pyDOE

Python framework for standard experimental design



pyDOE

Design of Experiments for Python

Overview Factorial Designs Response Surface Designs Randomized Designs

previous | next | index

pyDOE: The experimental design package for python

The pyDOE package is designed to help the **scientist, engineer, statistician,** etc., to construct appropriate **experimental designs**.

Hint

All available designs can be accessed after a simple import statement:

```
>>> from pyDOE import *
```

Capabilities

The package currently includes functions for creating designs for any number of factors:

- *Factorial Designs*
  1. *General Full-Factorial* (fullfact)
  2. *2-Level Full-Factorial* (ff2n)
  3. *2-Level Fractional-Factorial* (fracfact)
  4. *Plackett-Burman* (pbdesign)
- *Response-Surface Designs*
  1. *Box-Behnken* (bbdesign)
  2. *Central-Composite* (ccdesign)
- *Randomized Designs*
  1. *Latin-Hypercube* (lhs)

Table of contents

Section contents

Quick search

Go

Latin design

Window honors Ronald Fisher. Fisher's student, A. W. F. Edwards, designed this window for Caius College, Cambridge.

Halton sequences

[Halton, 1964]

- Used to generate points in $(0,1) \times (0,1)$
- Sequence that is constructed according to a deterministic method that uses a prime number as its base.

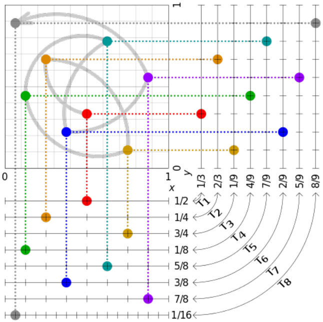
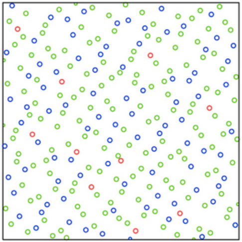


Figure source: Wikipedia

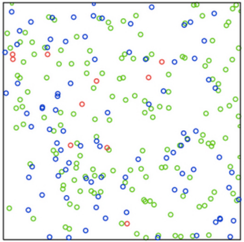
Halton sequences

[Halton, 1964]

Better coverage than random.



Halton



Random

Figure source: Wikipedia

Determinantal point processes

Kulesza and Taskar, [2012]

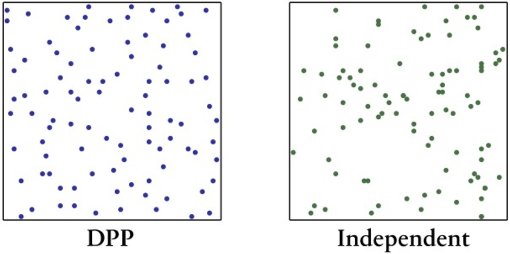
We say that X is a 'determinantal point process' on Λ with kernel K if it is a simple point process on Λ with a joint intensity or "correlation function" given by

$$\rho_n(x_1, \ldots, x_n) = \det(K(x_i, x_j)_{1 \leq i,j \leq n})$$

- Probability measures over subsets.
- Possible to characterise the samples in terms of quality and diversity.

Determinantal point processes

Kulesza and Taskar, [2012]



Key idea:

$$\mathcal{P}(i, j \in \boldsymbol{Y}) = \begin{vmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{vmatrix}$$
$$= K_{ii}K_{jj} - K_{ij}K_{ji}$$
$$= \mathcal{P}(i \in \boldsymbol{Y})\mathcal{P}(j \in \boldsymbol{Y}) - K_{ij}^2 .$$

Determinantal point processes

Kulesza and Taskar, [2012]

Why these ideas have been ignored for years?

- BO depends on its own parameters.
- Lack of software to apply these methods as a black optimzation boxes.
- Reduced scalability in dimensions and number of evaluations (this is still a problem).

Practical Bayesian Optimisation of Machine Learning Algorithms.
Snoek, Larochelle and Adams. NIPS 2012 (Spearmint)

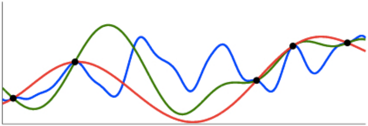
+

Other works of M. Osborne, P. Hennig, N. de Freitas, etc.

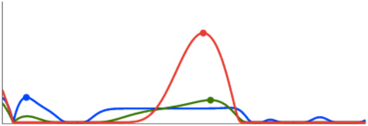
BO independent of the parameters of the GP.

[Snoek et al. 2012]

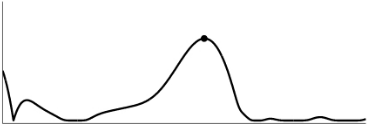
Integrate out across parameter values or location outputs.



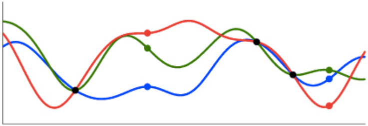
(a) Posterior samples under varying hyperparameters



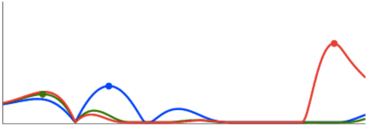
(b) Expected improvement under varying hyperparameters



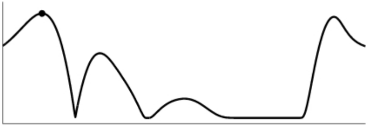
(c) Integrated expected improvement



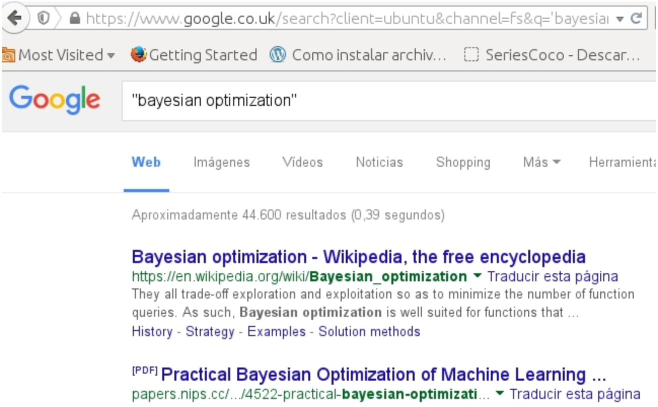
(a) Posterior samples after three data



(b) Expected improvement under three fantasies



(c) Expected improvement across fantasies



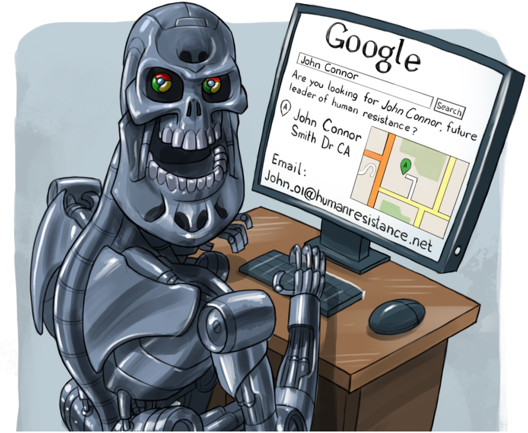
- Hot topic in Machine Learning.
- The BO workshop at NIPS is well stablished and it is a mini-conference itself.

Bayesian optimization now

What has made BO so popular is that by first time it has allowed to use Machine Learning algorithms without any human intervention.

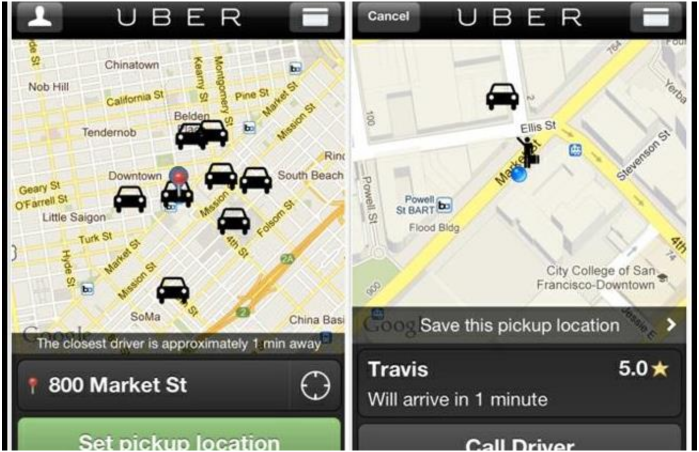
BO takes to human out of the loop!

BO takes to human out of the loop



BO in industry: Twitter

BO in industry: Uber

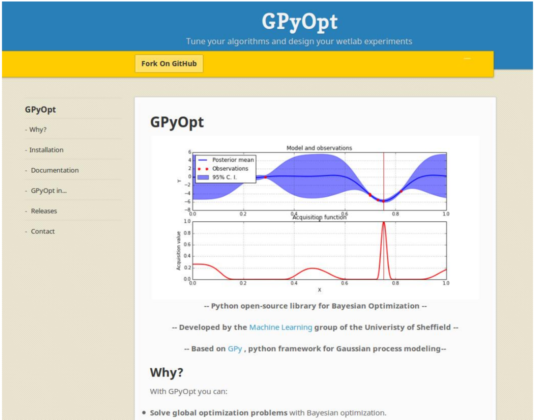


Available software

- Spearmint (https://github.com/HIPS/Spearmint).
- BayesOpt (http://rmcantin.bitbucket.org/html/).
- pybo (https://github.com/mwhoffman/pybo).
- robo (https://github.com/automl/RoBO).

Open Software: GPyOpt

http://sheffieldml.github.io/GPyOpt/



We will use it in the lab session

GPyOpt

- Python module for BO.
- Based on GPy. All functionalities available.
- Sparse GPs, Multi-output GPs, several likelihoods, etc.
- Parallel optimization.

GPyOpt: methods of use

Modular BO k = GPy.kern.RBF(1)

BO = BayesianOptimization(f=f, bounds=b, acquisition='EI', kernel=k)

BO.run_optimization(max_iter)

Automatic ML

param = GPyOpt.methods.autoTune(objective, bounds)

Use GPyOpt using the same interface as Spearmint

config.json + problem.py

Extensions of Bayesian Optimization (tomorrow!)

- Multi-task Bayesian optimization [Wersky et all., 2013].
- Bayesian optimization for high dimensional problems [Wang et al., 2013].
- Non-myopic methods [Osborne, 2010].
- Discrete domains (armed bandits) [Srinivas et al., 2010].
- Parallel approaches [Chevalier and Ginsbourger 2012].
- Conditional parameter spaces [Swersky et al. 2013].
- Applications to robotics, molecule design, etc.

Projects

- Work in groups of 2-3 people.
- Find and interesting function to optimize: computer model, physical experiment etc.
- Be original!
- Use BO principles to optimize the function.
- Write a small presentation/demonstration for the rest of the gorup (10 mins!)