# Introduction to Bayesian Optimization

**Javier González**

Gaussian process summer school, September, 2107
@Sheffield University

# Big picture

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

We are interested on automation:

- Automatic model configuration.
- Automate the design of physical experiments.

# Big picture

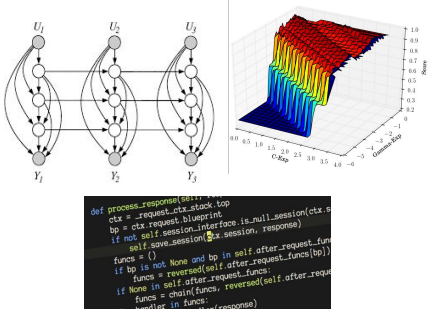
*"Civilization advances by extending the number of important operations which we can perform without thinking of them."*
*(Alfred North Whitehead)*

We are interested on automation:

- Automatic model configuration.
- Automate the design of physical experiments.

# 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**

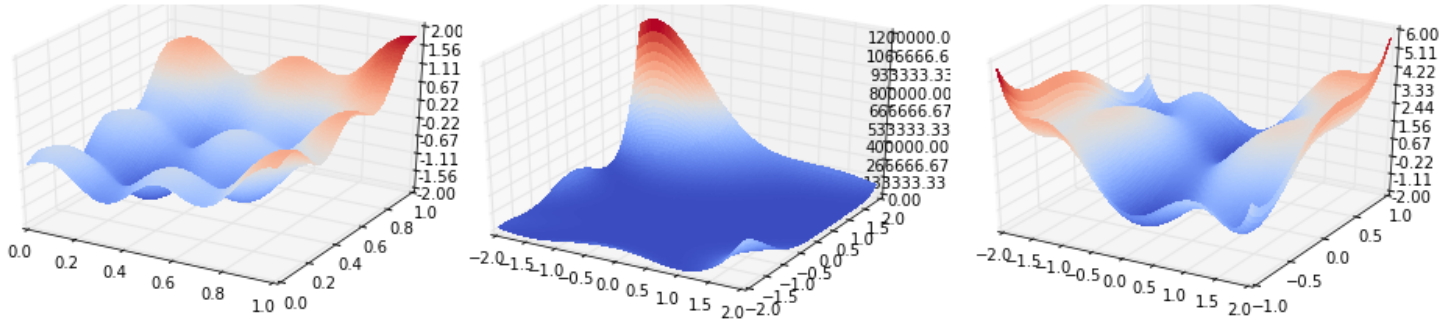
Data visualisation, Sequential experimentation

**Pipeline improvement / Interaction with environment**

# Global optimization

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

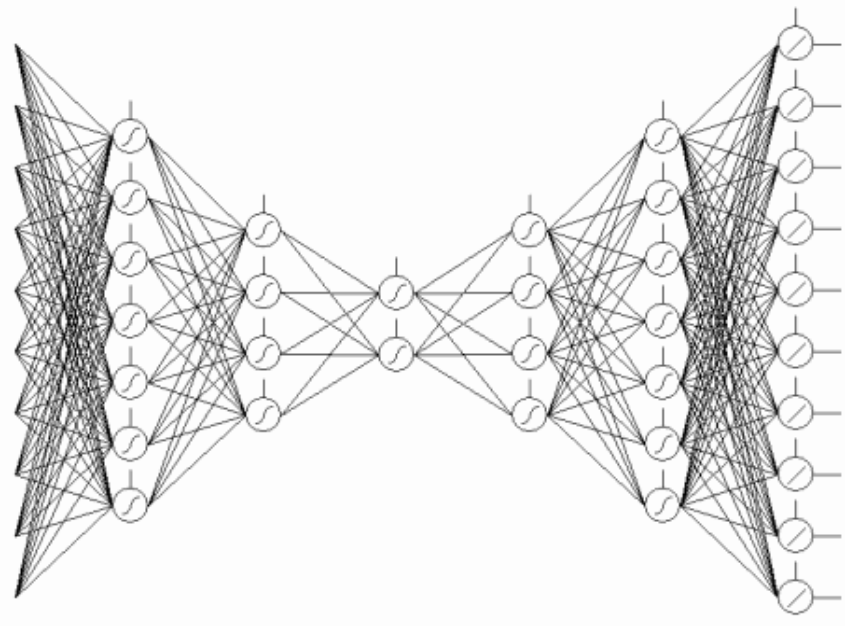
$$x_M = \arg\min_{x \in \mathcal{X}} f(x).$$



- $f$ 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.

# Expensive functions, who doesn't have one?

## Active Path Finding


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?

Many other problems:

- Robotics, control, reinforcement learning.
- Scheduling, planning.
- Compilers, hardware, software.
- Industrial design.
- Intractable likelihoods.

# What to do?

**Option 1**: Use previous knowledge

**Option 2**: Grid search? $f$ is L-Lipschitz continuous and we are in a noise-free domain. To propose $x_{M,n}$ 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!

**Option 3**: We can sample the space uniformly [Bergstra and Bengio 2012]

Can we do better?

# What to do?

**Option 1**: Use previous knowledge

**Option 2**: Grid search? $f$ is L-Lipschitz continuous and we are in a noise-free domain. To propose $x_{M,n}$ 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!

**Option 3**: We can sample the space uniformly [Bergstra and Bengio 2012]

Can we do better?

# What to do?

**Option 1**: Use previous knowledge

**Option 2**: Grid search? $f$ is L-Lipschitz continuous and we are in a noise-free domain. To propose $x_{M,n}$ 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!

**Option 3**: We can sample the space uniformly [Bergstra and Bengio 2012]

Can we do better?

# What to do?

**Option 1**: Use previous knowledge

**Option 2**: Grid search? $f$ is L-Lipschitz continuous and we are in a noise-free domain. To propose $x_{M,n}$ 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!

**Option 3**: We can sample the space uniformly [Bergstra and Bengio 2012]

Can we do better?

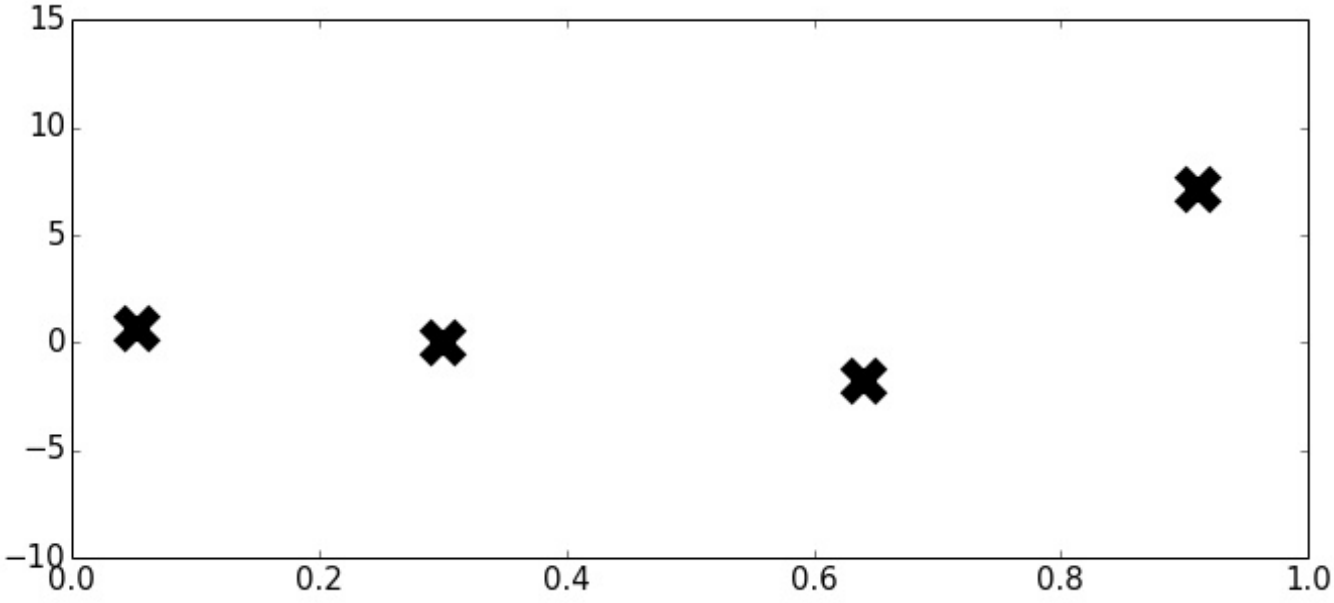
# Problem for the audience of this tutorial!

You know that:

- 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

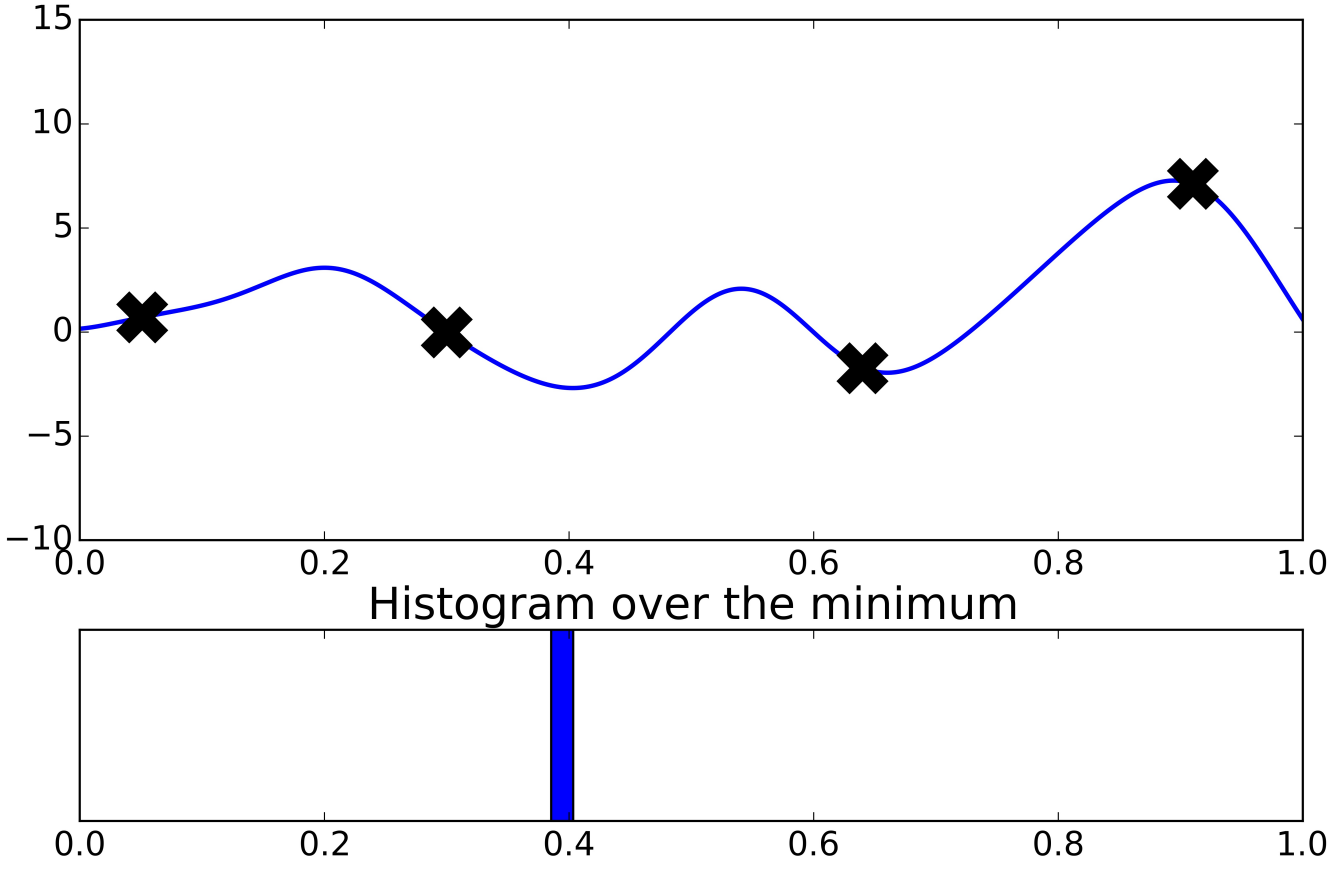
## We have a few function evaluations




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

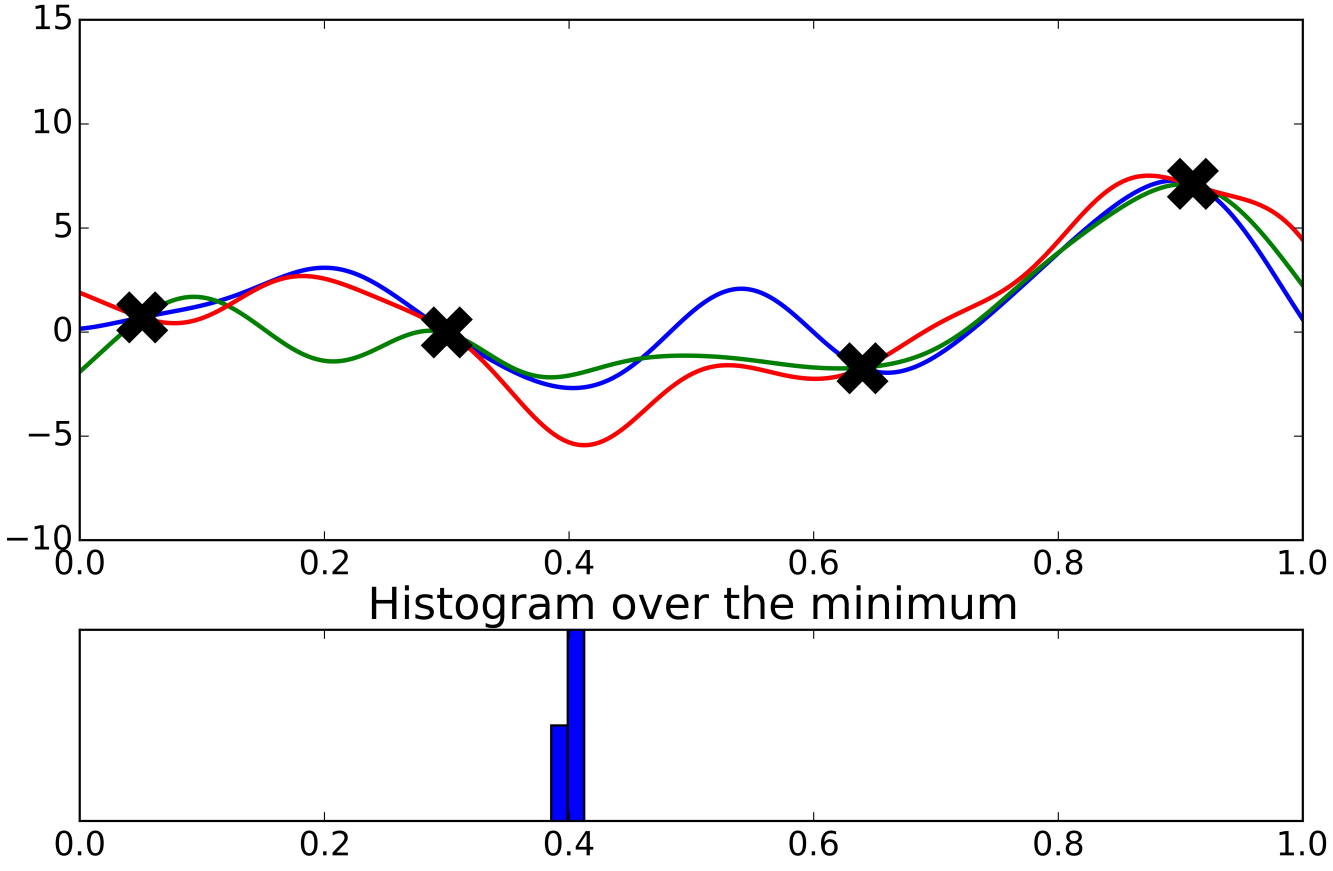
# Intuitive solution

## One curve



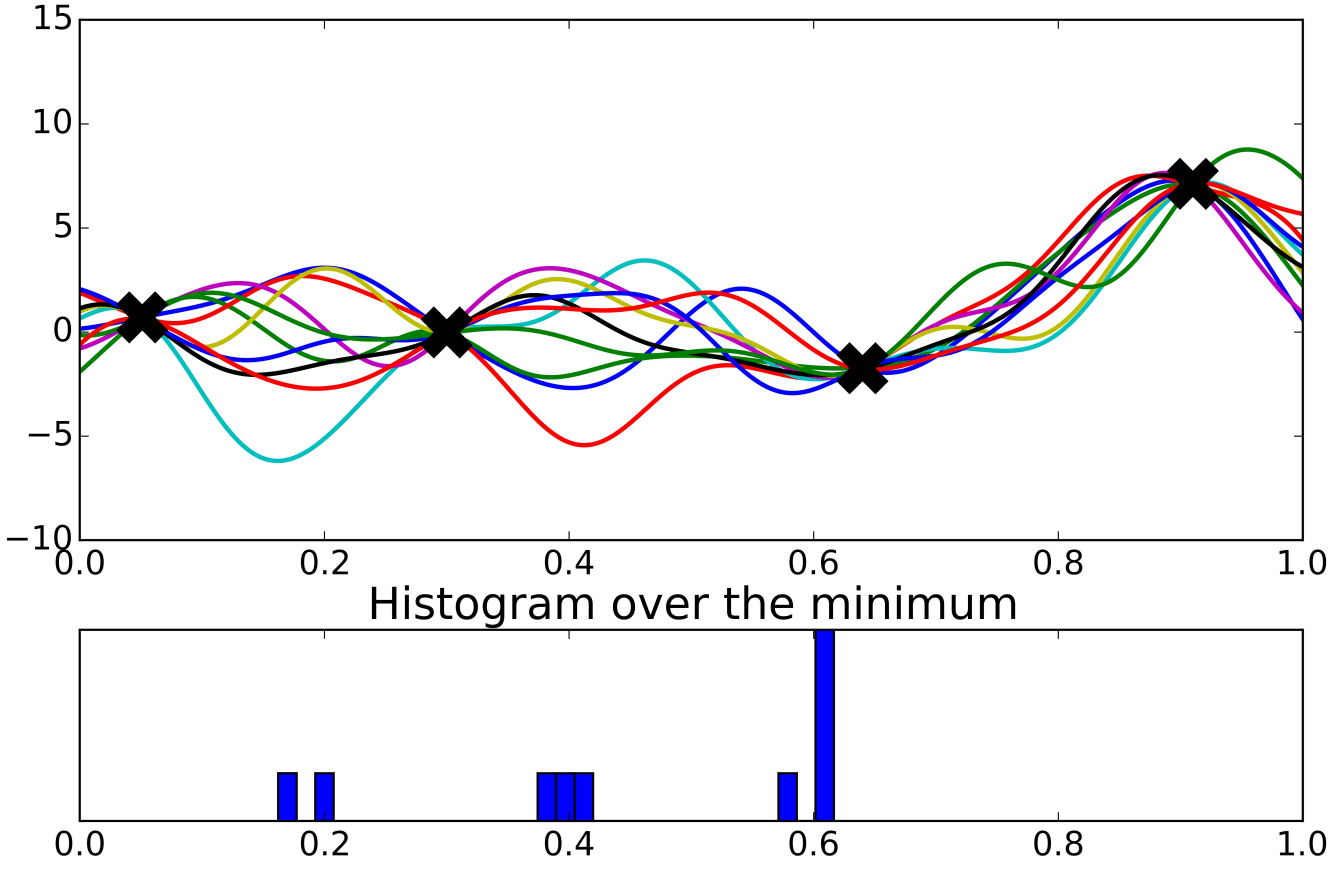
# Intuitive solution

## Three curves



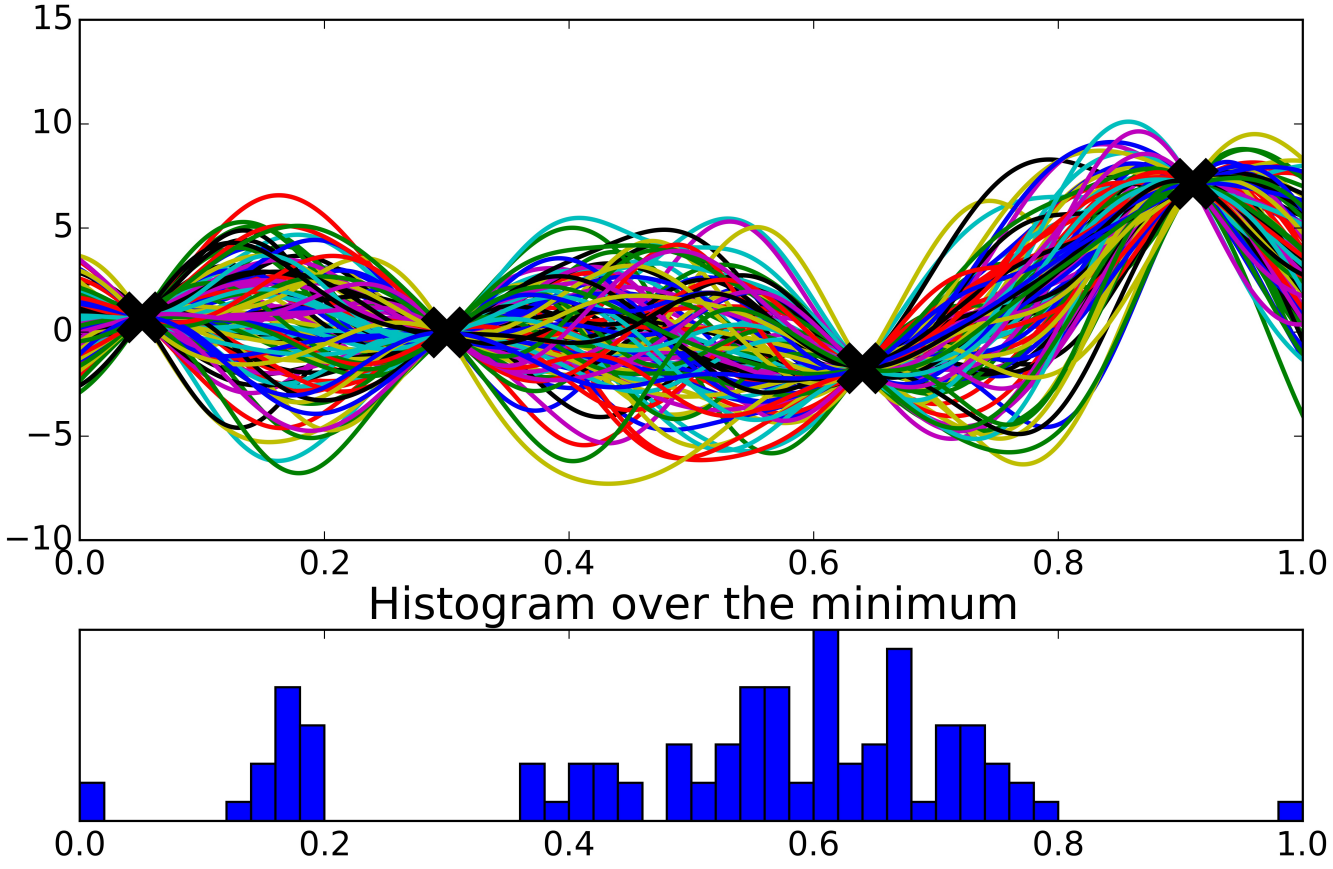
# Intuitive solution

## Ten curves



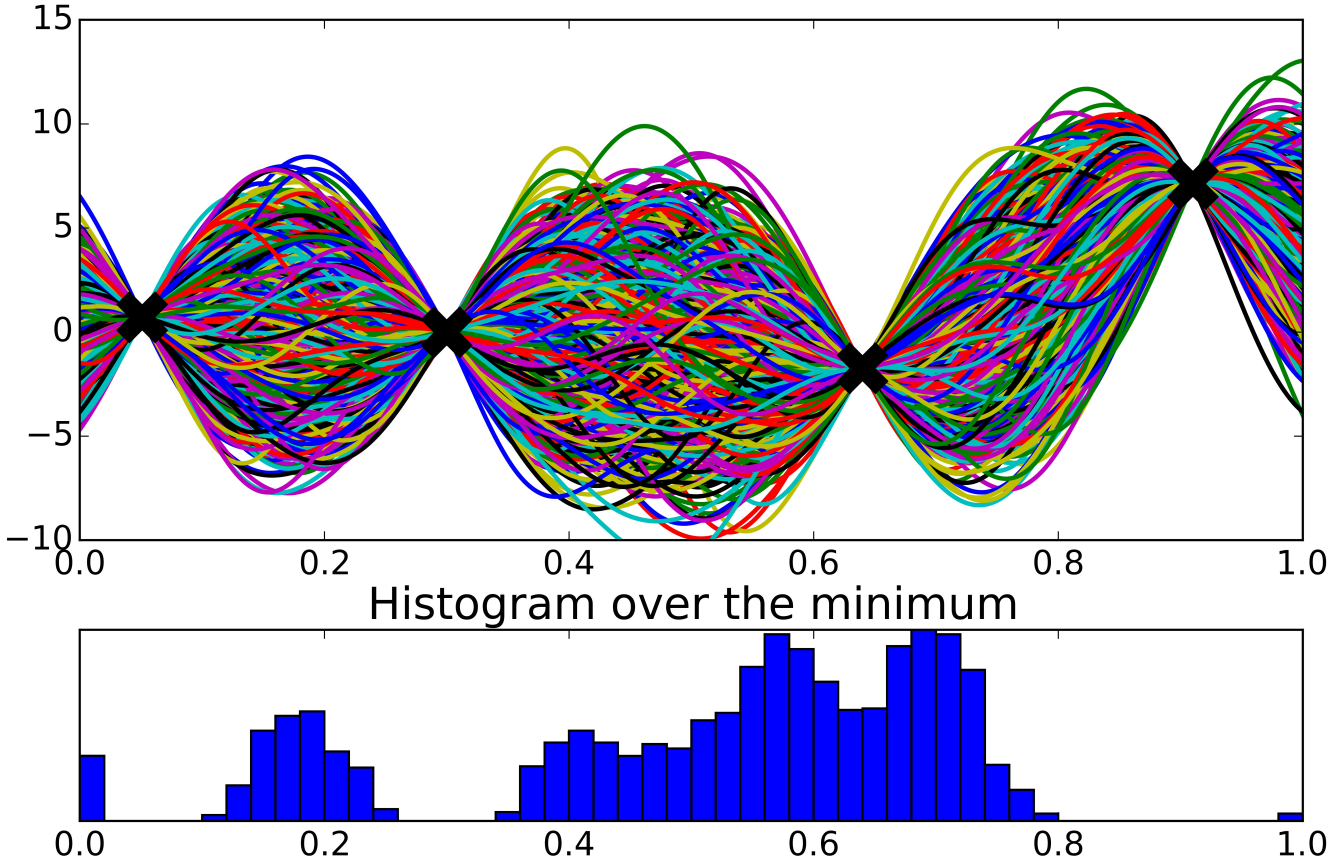
# Intuitive solution

## Hundred curves



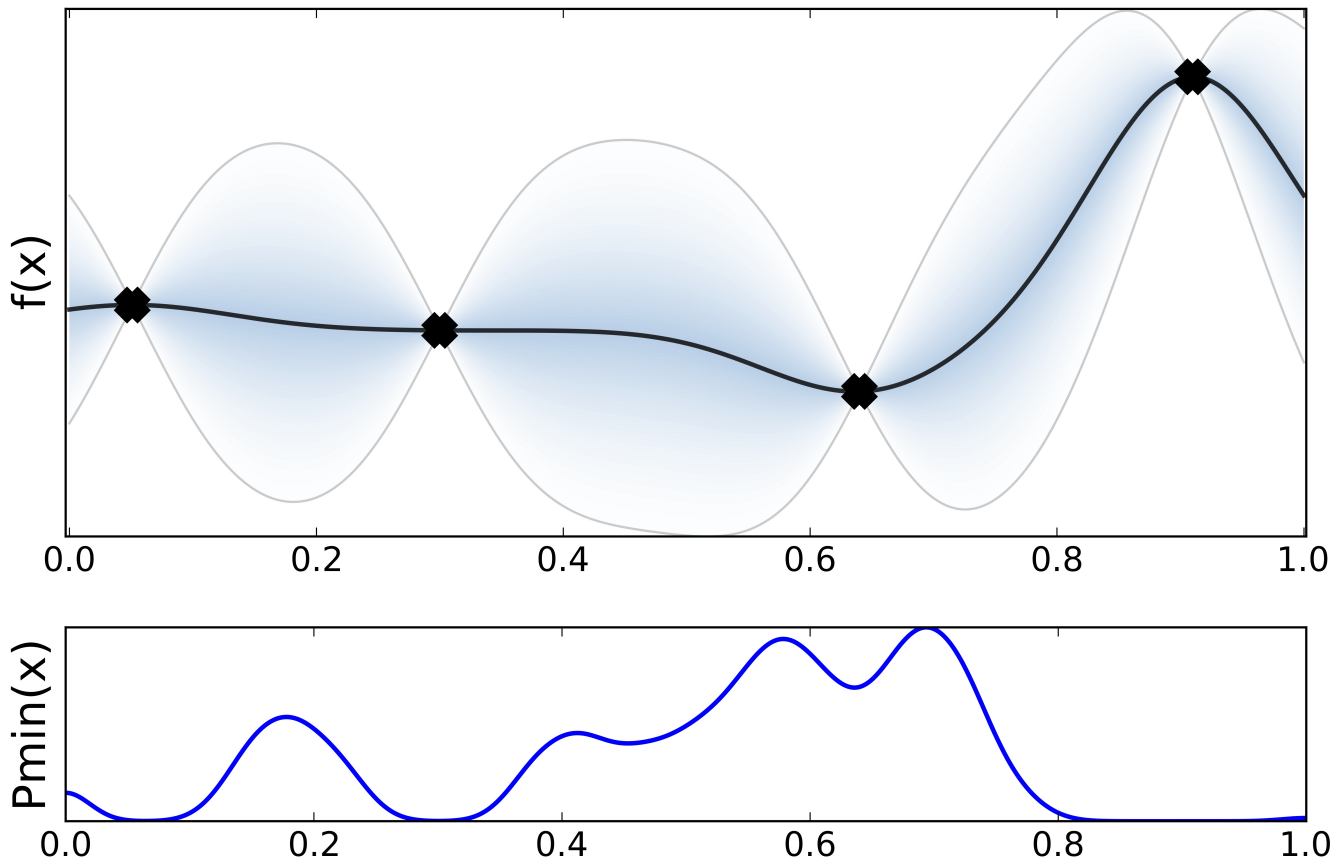
# Intuitive solution

## 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: Making informed decisions*

# Utility functions

The utility should represent our design goal:.

1. **Active Learning and experimental design**: reduce the uncertainty in the model (prediction or hyper-parameters).

2. **Optimization**: Minimize the loss in a sequence $x_1, \ldots, x_n$

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

(1) does to a lot exploration whereas (2) encourages exploitation about the minimum of $f$.

# 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*.

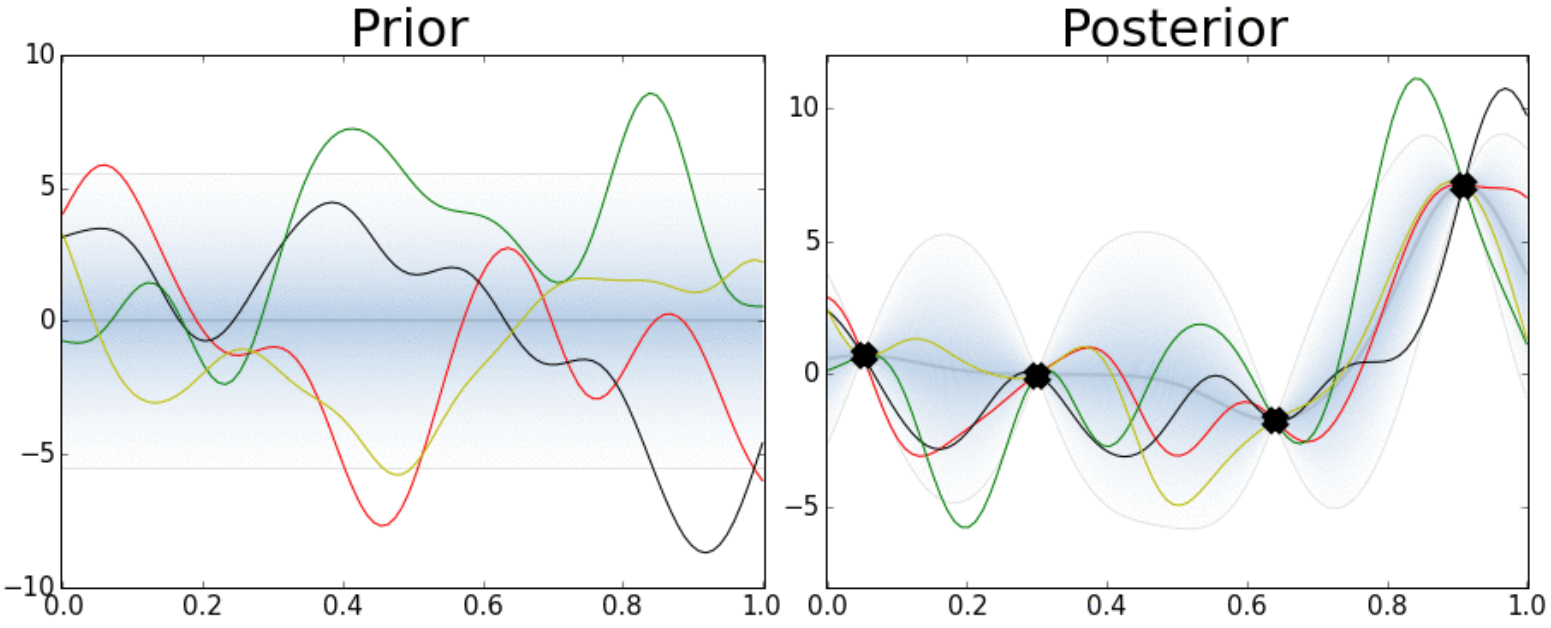
4. Augment the data.

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.

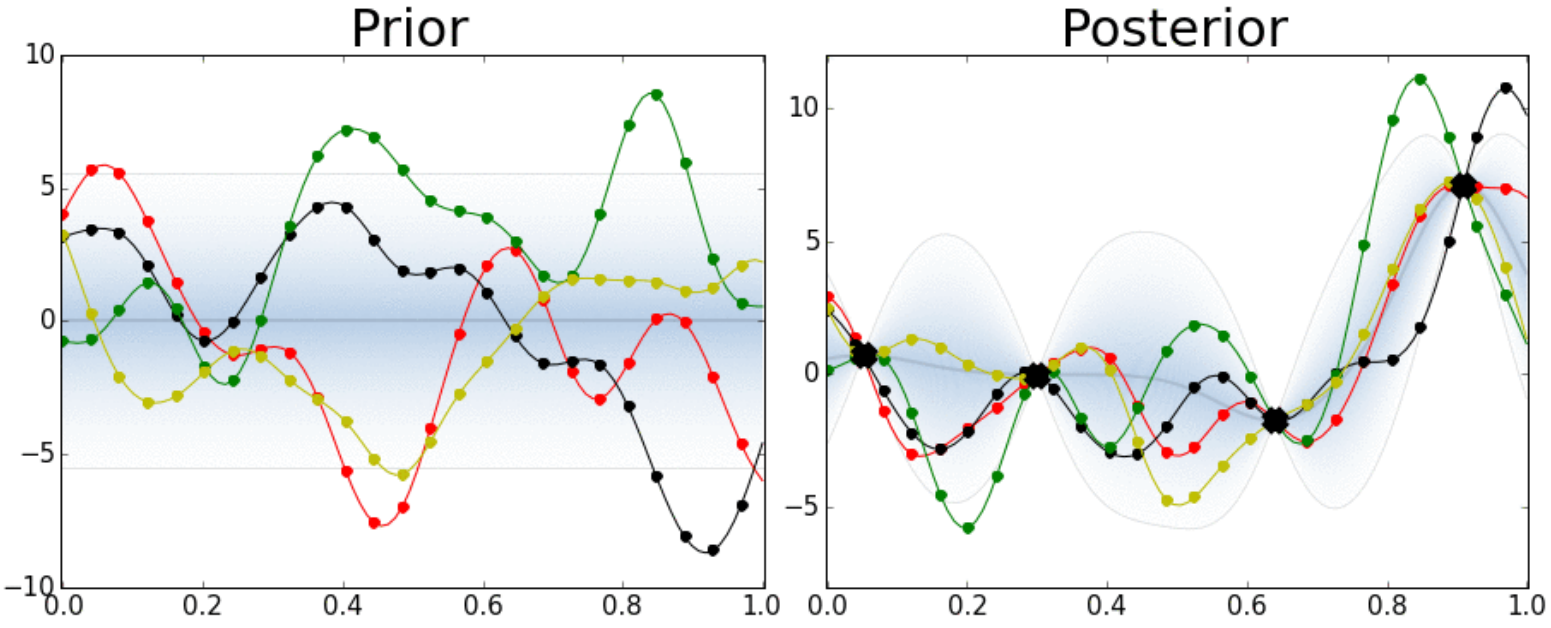



- Model $f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$ is determined by the *mean function* $m(x)$ and *covariance function* $k(x, x'; \theta)$.

# 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* $k(x, x'; \theta)$.

# Exploration vs. exploitation

[Borji and Itti, 2013]

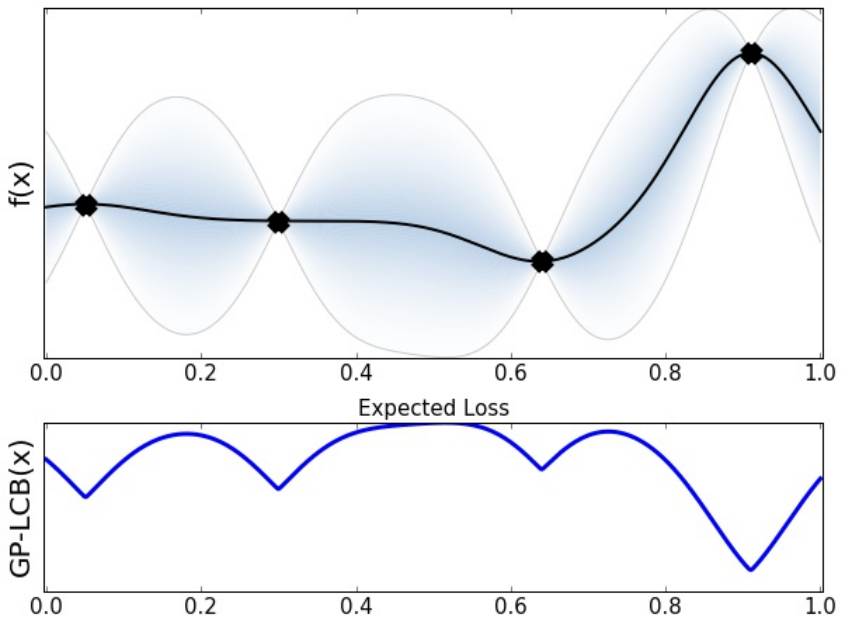

Bayesian optimization explains human active search

# GP Upper (lower) Confidence Band

[Srinivas et al., 2010]

Direct balance between exploration and exploitation:

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



# 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'.

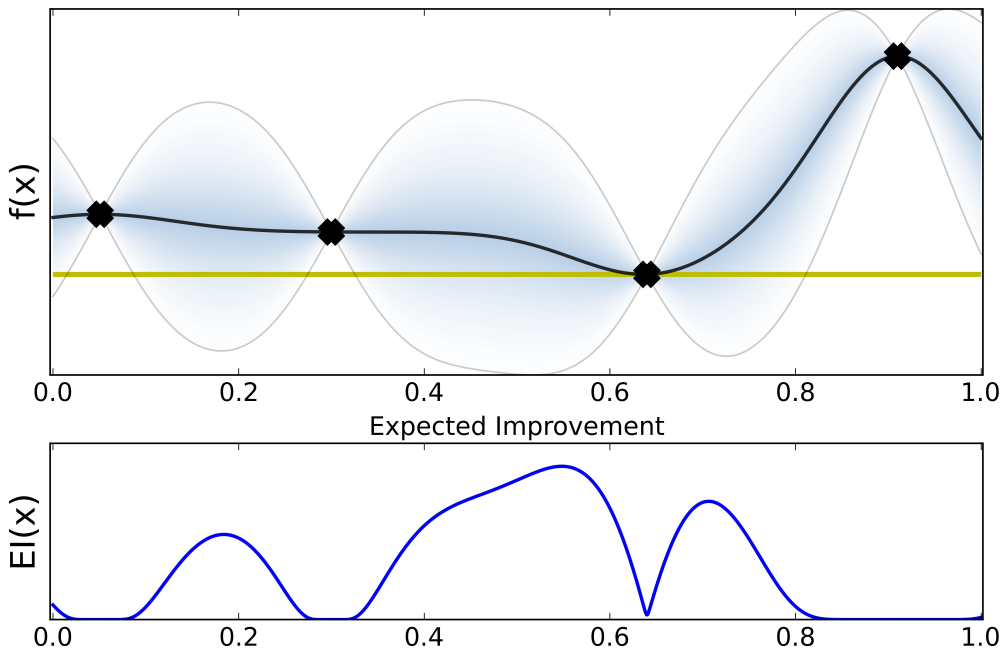
**Theorem 1** *Let* $\delta \in (0,1)$ *and* $\beta_t = 2\log(|D|t^2\pi^2/6\delta)$. *Running GP-UCB with* $\beta_t$ *for a sample* $f$ *of a GP with mean function zero and covariance function* $k(\boldsymbol{x}, \boldsymbol{x}')$, *we obtain a regret bound of* $\mathcal{O}^*(\sqrt{T\gamma_T \log|D|})$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

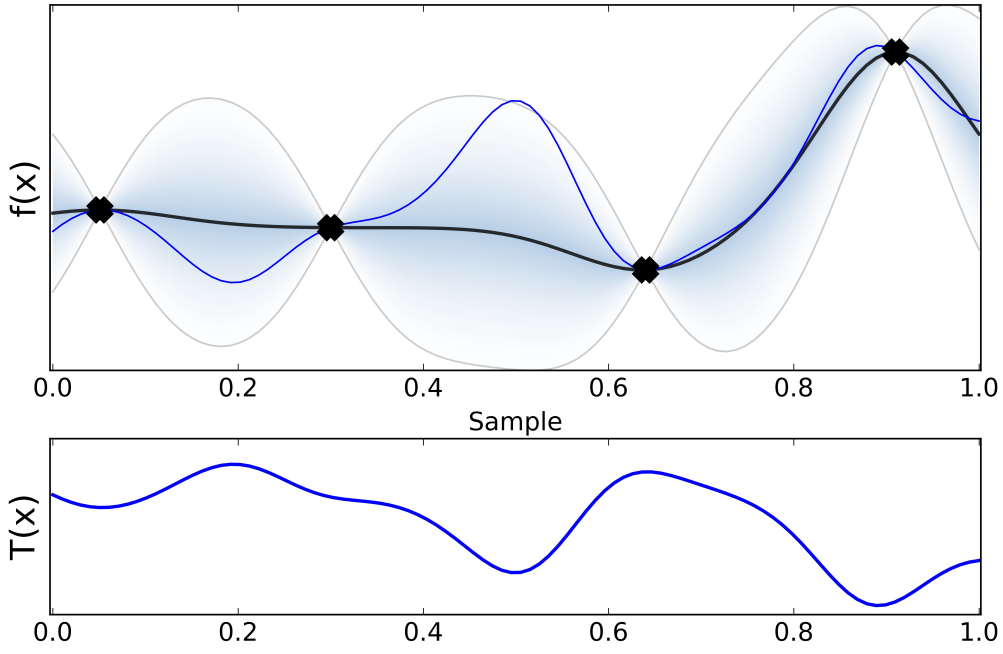
$$\gamma(x) = \frac{f(x_{best}) - \mu(\mathbf{x}; \theta, \mathcal{D}) + \psi}{\sigma(\mathbf{x}; \theta, \mathcal{D})}.$$

# Thompson sampling

## Probability matching [Rahimi and B. Recht, 2007]

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

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



# Thompson 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 $k$

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

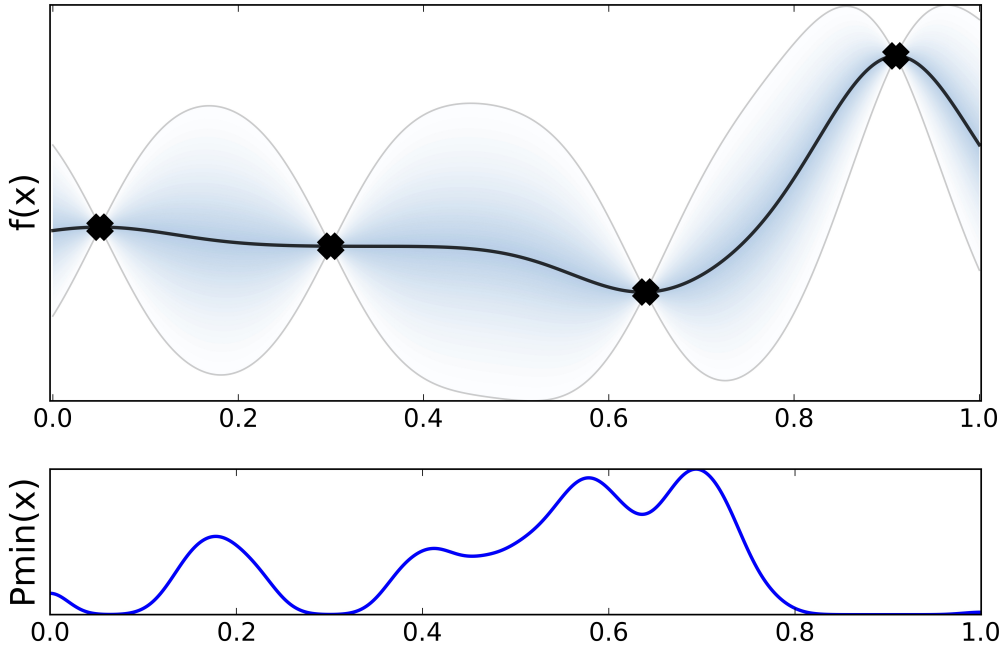
With sampling and this lemma (taking $p(w) = s(\omega)/\nu$ and $b \sim \mathcal{U}[0, 2\pi]$) we can construct a feature based approximation for sample paths of the GP.

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

# 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

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

Uses the distribution of the minimum

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

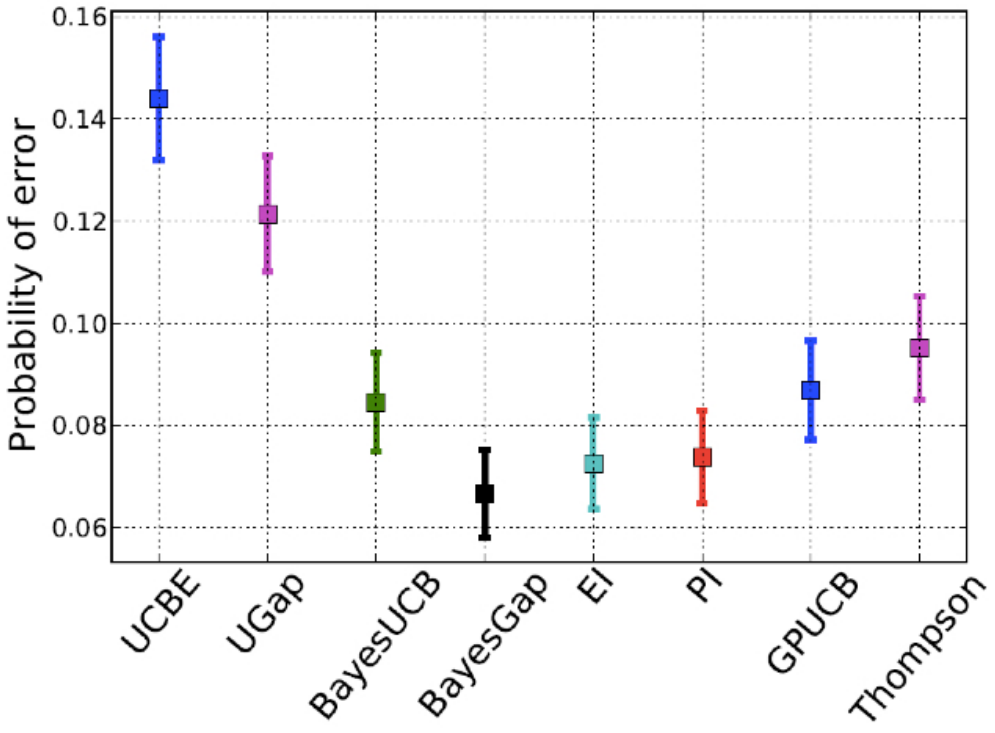
where $\theta$ is the Heaviside's step function. No closed form!

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

# The choice of utility matters

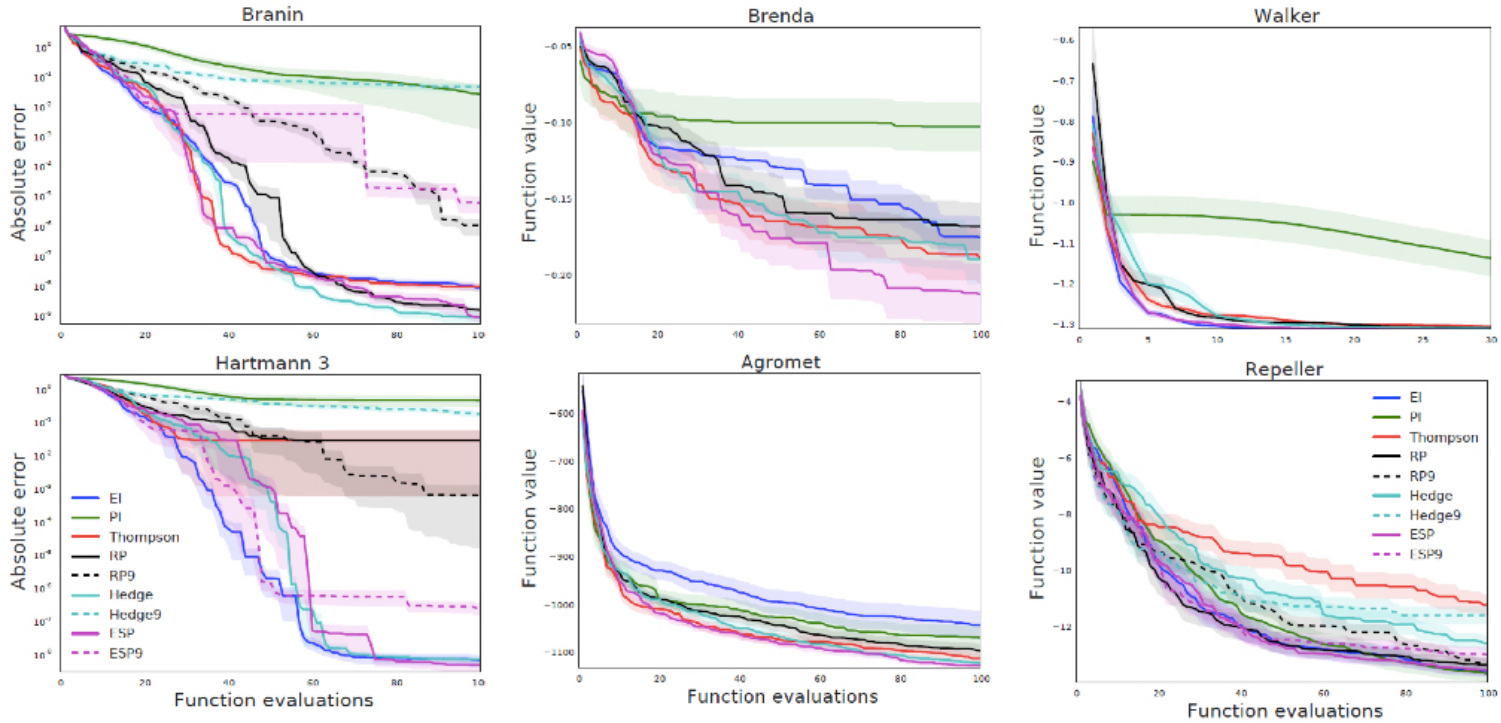
[Hoffman, Shahriari and de Freitas, 2013]

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



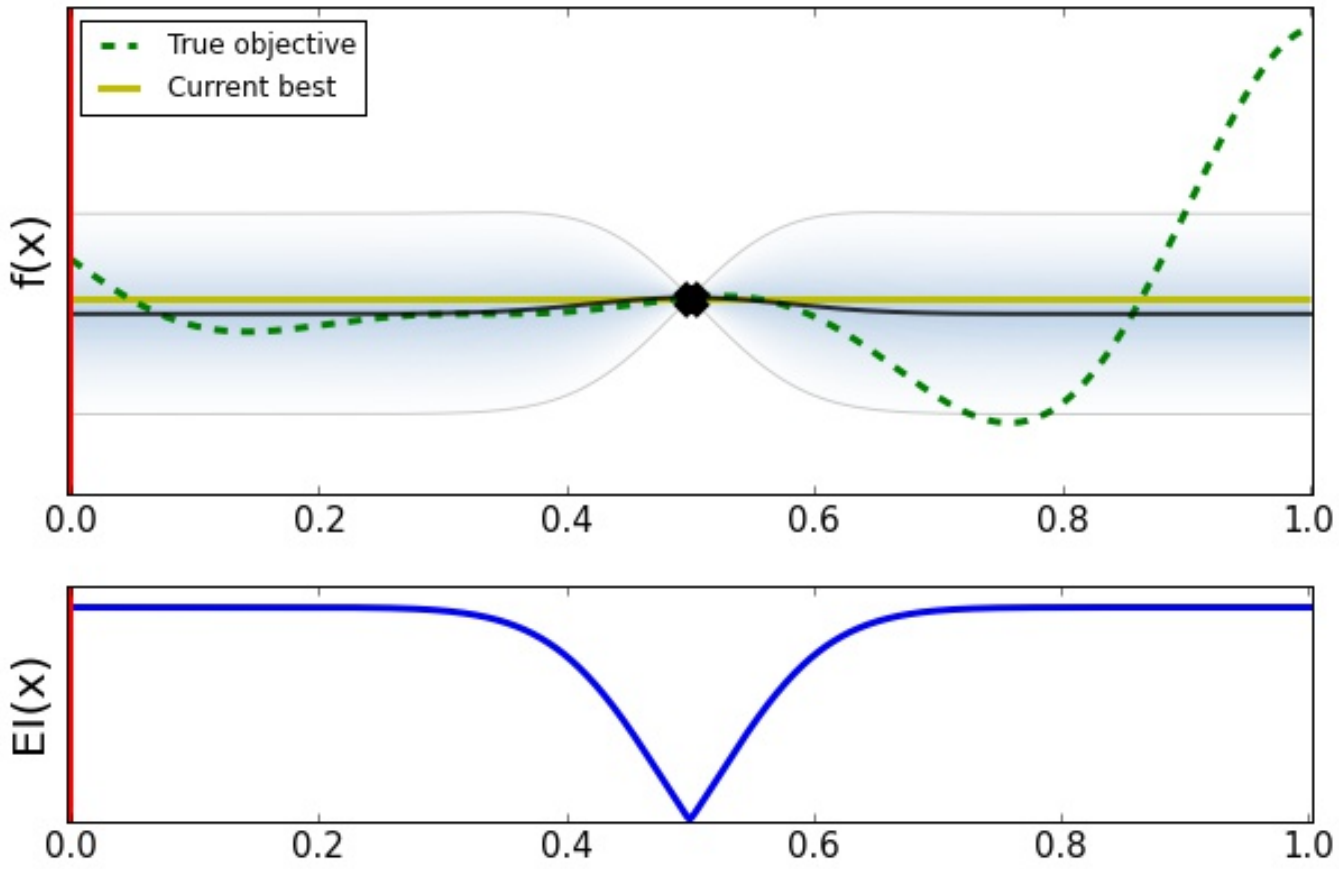
# The choice of utility in practice

[Hoffman, Shahriari and de Freitas, 2013]

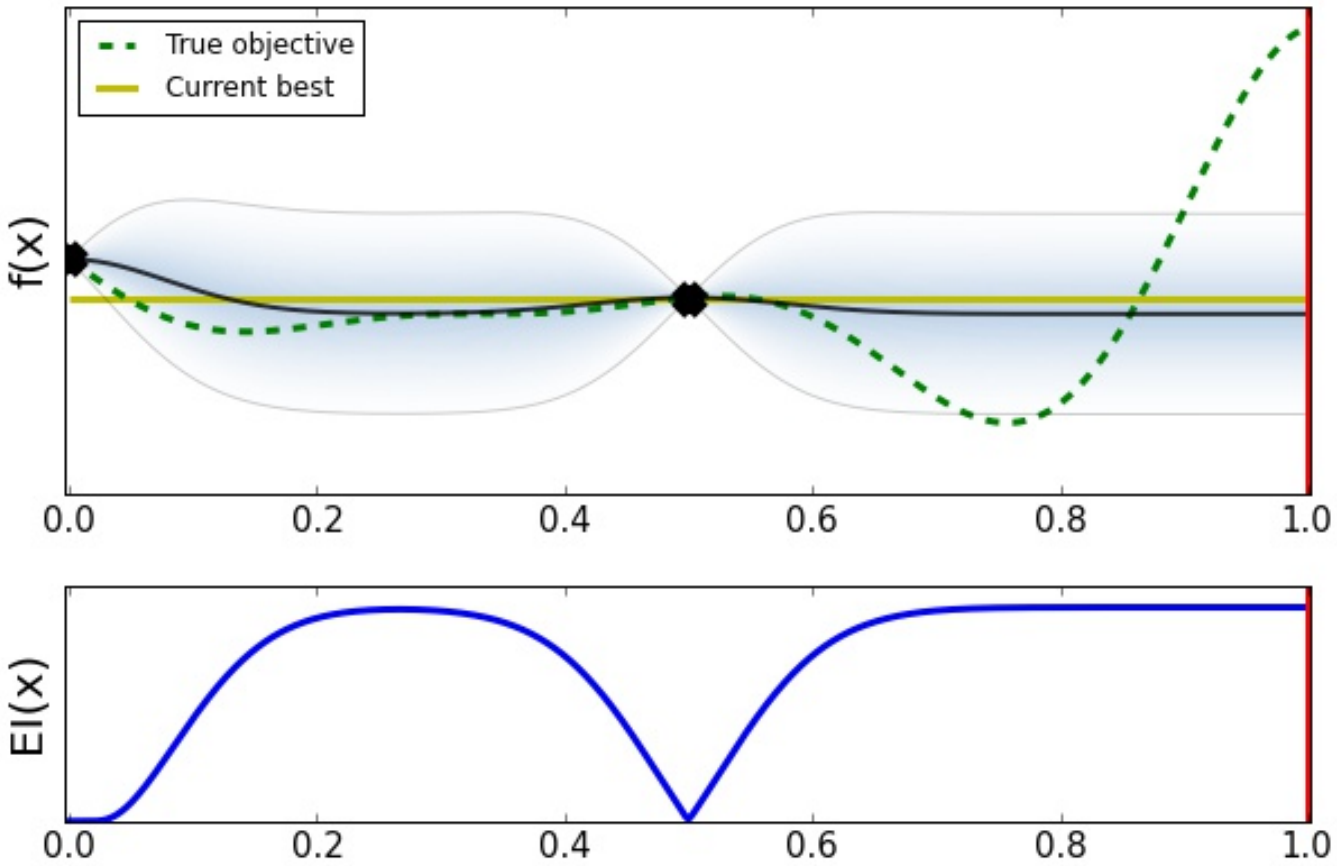



The best utility depends on the problem and the level of exploration/exploitation required.

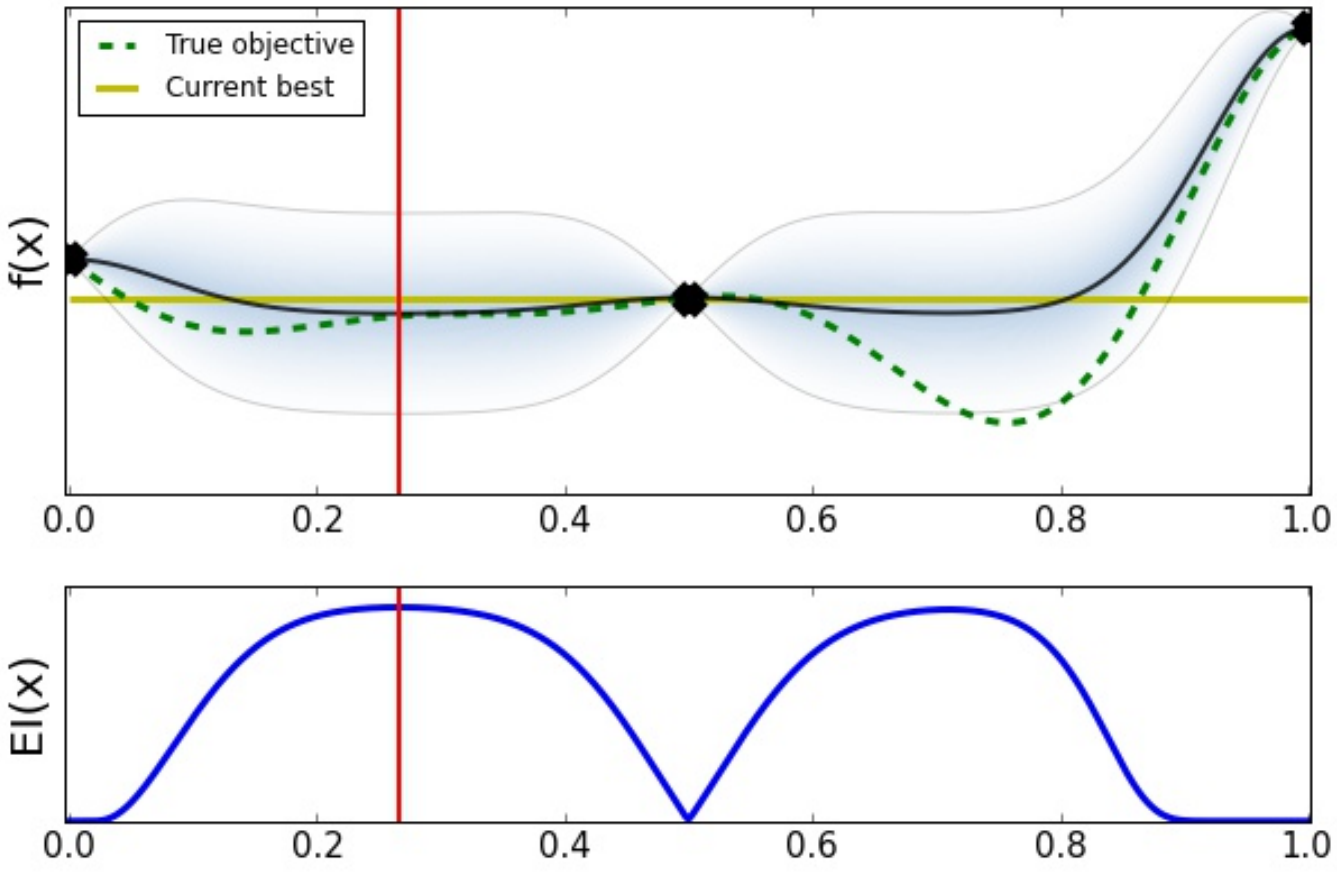
# Illustration of BO



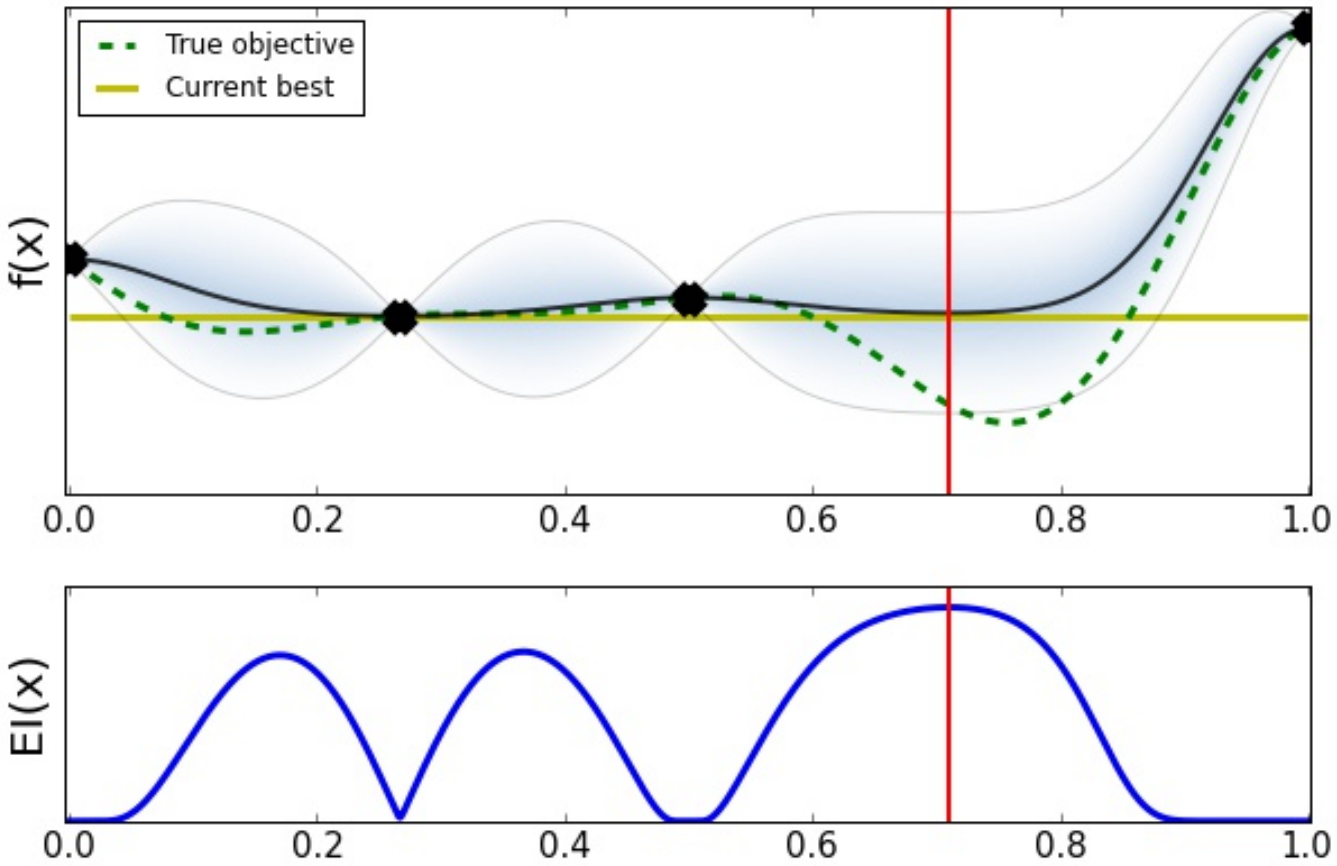
# Illustration of BO



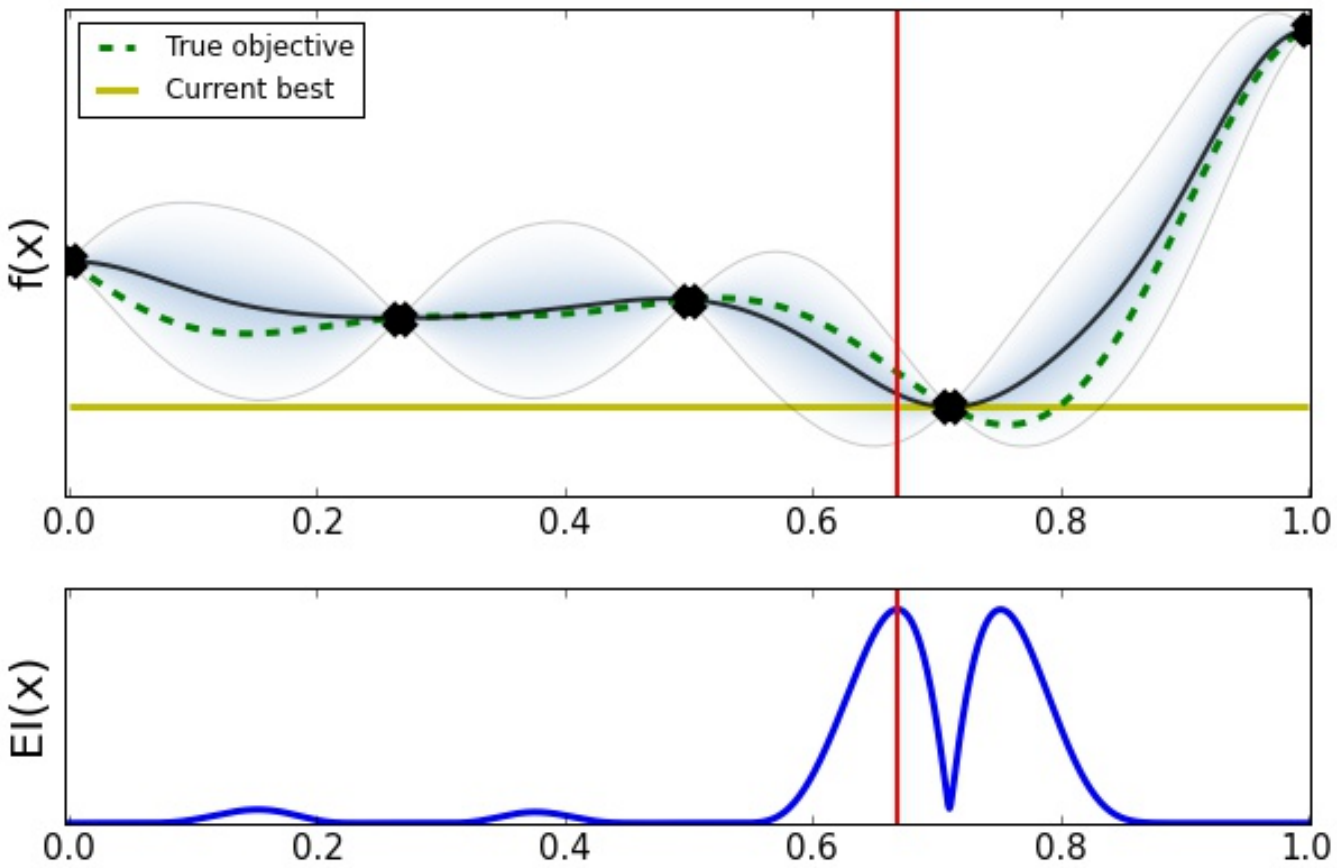
# Illustration of BO



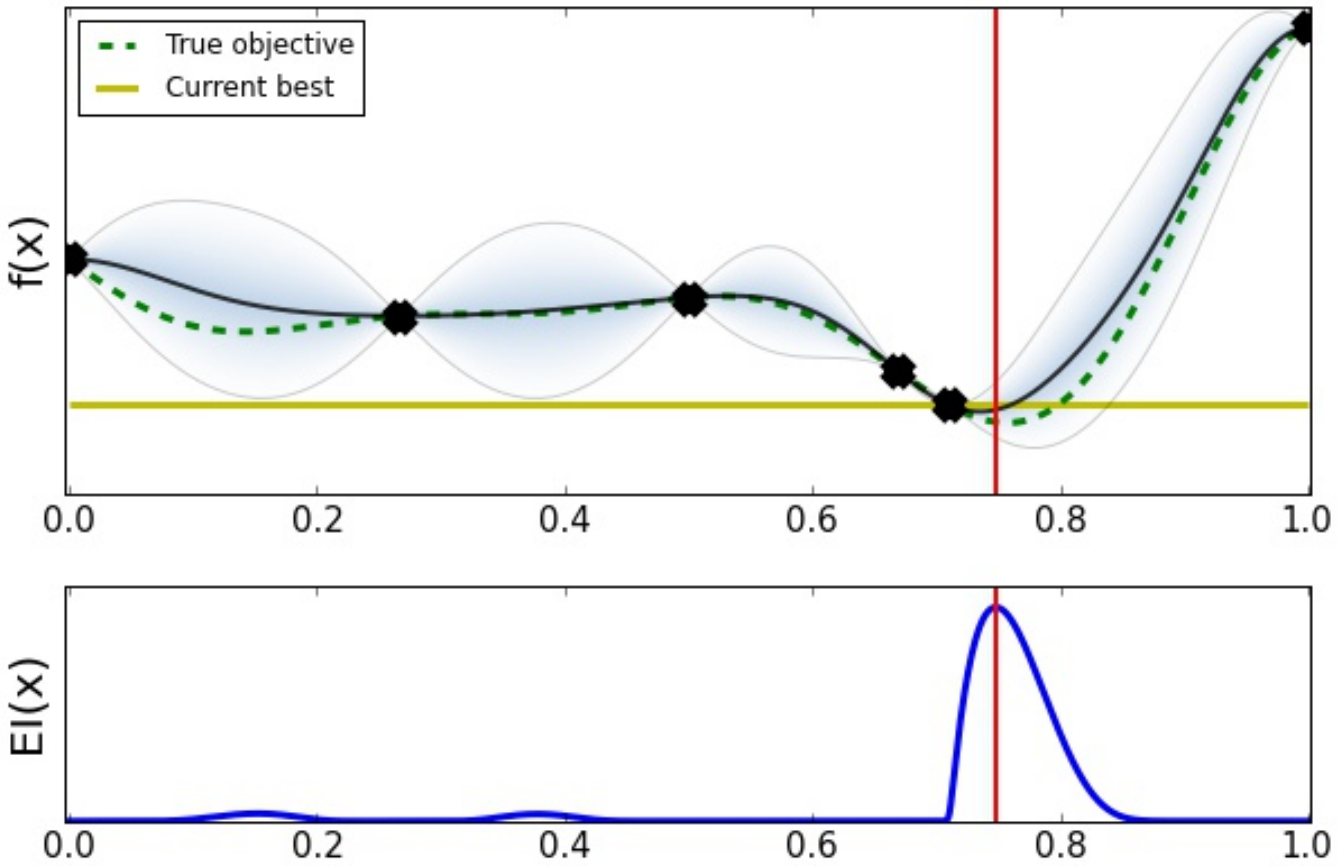
# Illustration of BO



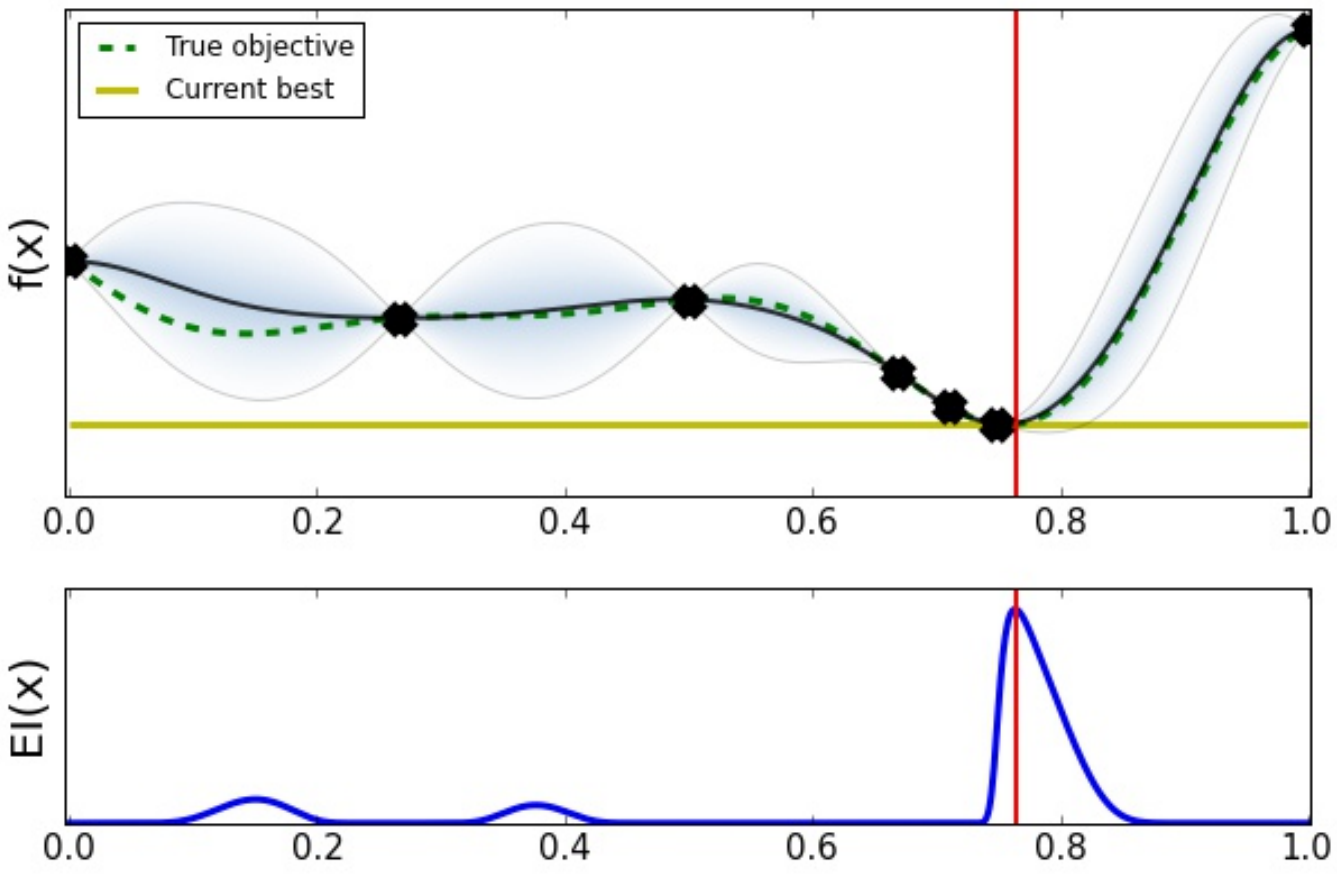
# Illustration of BO



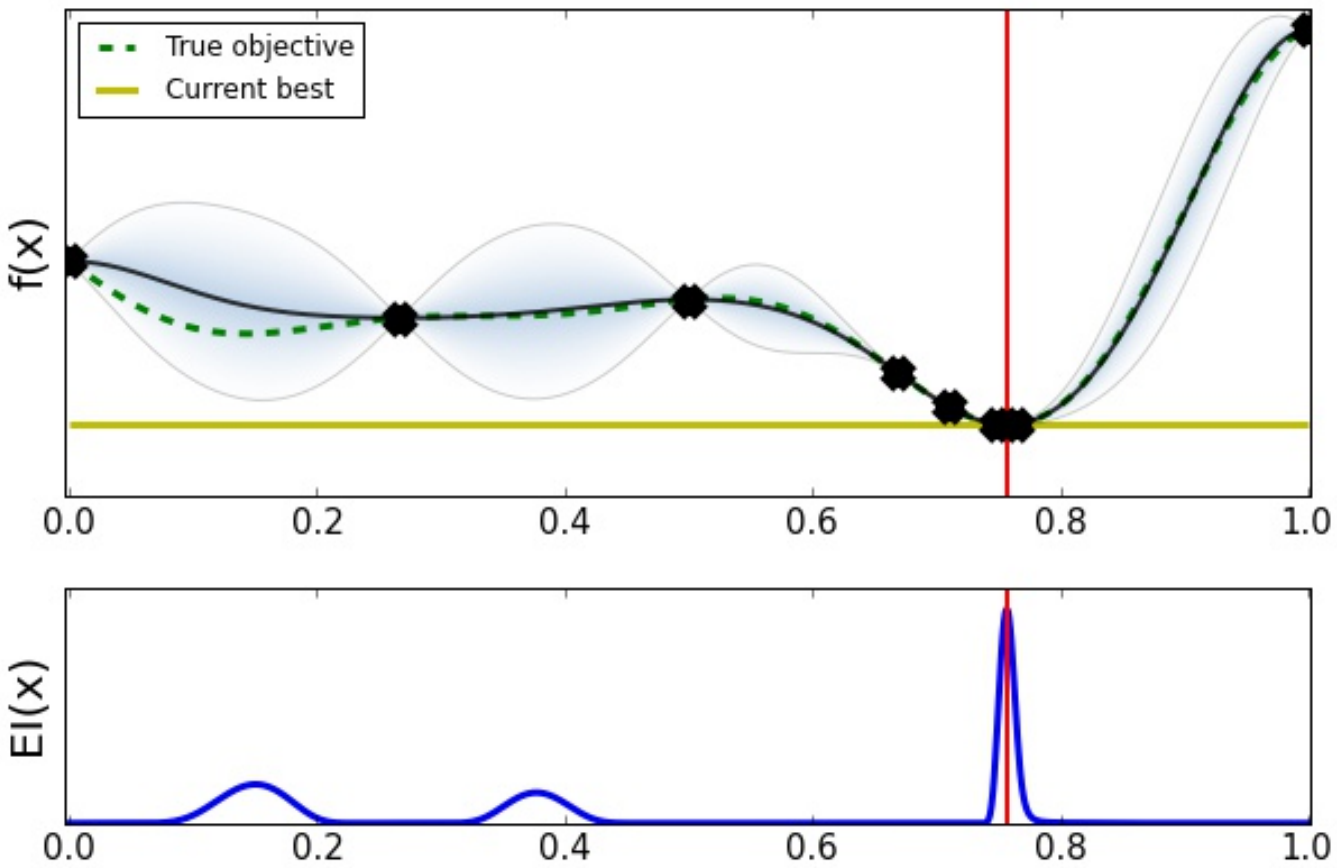
# Illustration of BO



# Illustration of BO



# Illustration of BO



# Bayesian Optimization

## As a 'mapping' between two problems

BO is an strategy to transform the problem

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

*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 $x_{n+1}$.

# Secrets of Bayesian optimization...

Be aware that:

- The model matters a lot! Be sure that you have it right (sample hyper-parameters, prior knowledge, etc.)
- Optimizing the acquisition can be hard (multimodal) but you can use standard techniques.
- Be aware of the input dimension. Up to 10 dimensions is OK, If you have more probably you'll need to impose some structure in the problem.

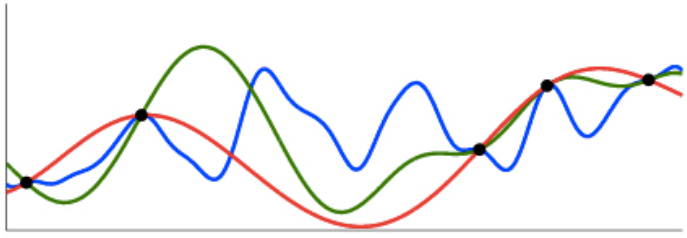
# Main issues

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

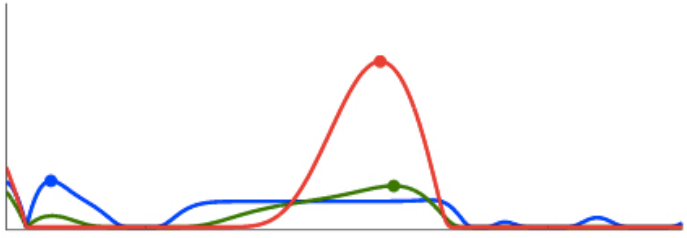
# BO independent of the parameters of the GP.

[Snoek et al. 2012]

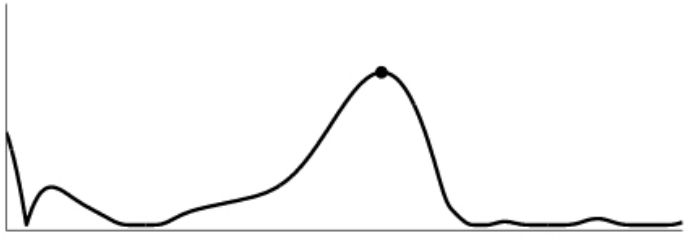
Integrate out across parameter values or location outputs.



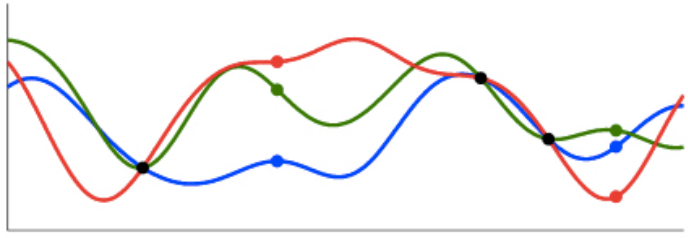
(a) Posterior samples under varying hyperparameters



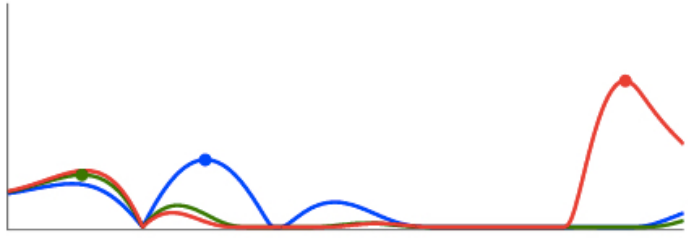
(b) Expected improvement under varying hyperparameters



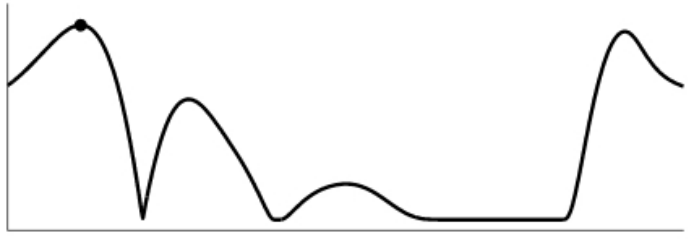
(c) Integrated expected improvement



(a) Posterior samples after three data



(b) Expected improvement under three fantasies



(c) Expected improvement across fantasies

# How to initialise the model?

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

# Determinantal point processes

Kulesza and Taskar, [2012]

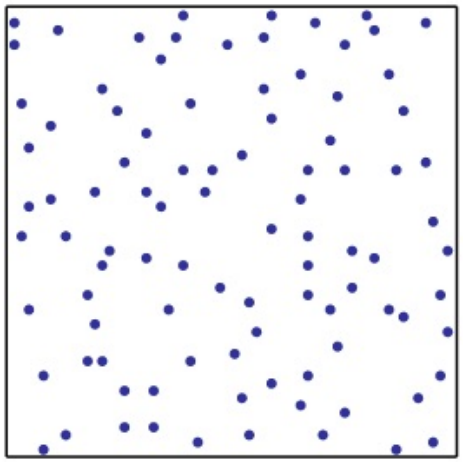
We say that $X$ is a 'determinantal point process' on $\Lambda$ with kernel $K$ if it is a simple point process on $\Lambda$ with a joint intensity or 'correlation function' given by

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

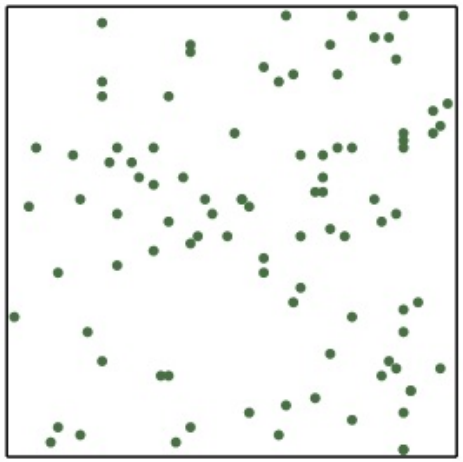
- Probability measures over subsets.
- Possible to characterise the samples in terms of quality and diversity.

# Determinantal point processes

Kulesza and Taskar, [2012]



DPP



Independent

Key idea:

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

# Determinantal point processes

Kulesza and Taskar, [2012]

# Methods to optimise the acquisition function

This may not be easy.

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

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

# Some advanced topics

- Multi-task Bayesian optimization
- Early stopping
- Parallel Bayesian optimization.
- Non myopic menthods
- Conditional dependencies.
- Preferential optimization.

# Multi-task Bayesian Optimization

[Wersky et al., 2013]

Two types of problems:

1. Multiple, and conflicting objectives: design an engine more powerful but more efficient.

2. The objective is very expensive, but we have access to another cheaper and correlated one.

# Multi-task Bayesian Optimization

[Wersky et al., 2013]

- We want to optimise an objective that it is very expensive to evaluate but we have access to another function, correlated with objective, that is cheaper to evaluate.

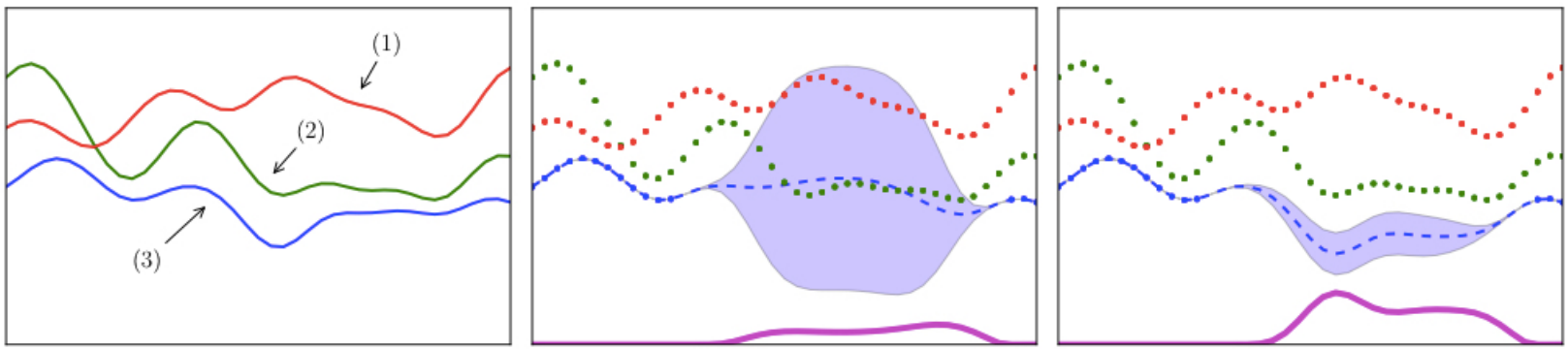
- The idea is to use the correlation among the function to improve the optimization.

Multi-output Gaussian process

$$\tilde{k}(x, x') = \mathbf{B} \otimes k(x, x')$$

# Multi-task Bayesian Optimization

[Wersky et al., 2013]




(a) Multi-task GP sample functions (b) Independent GP predictions (c) Multi-task GP predictions

- Correlation among tasks reduces global uncertainty.
- The choice (acquisition) changes.

# Multi-task Bayesian Optimization

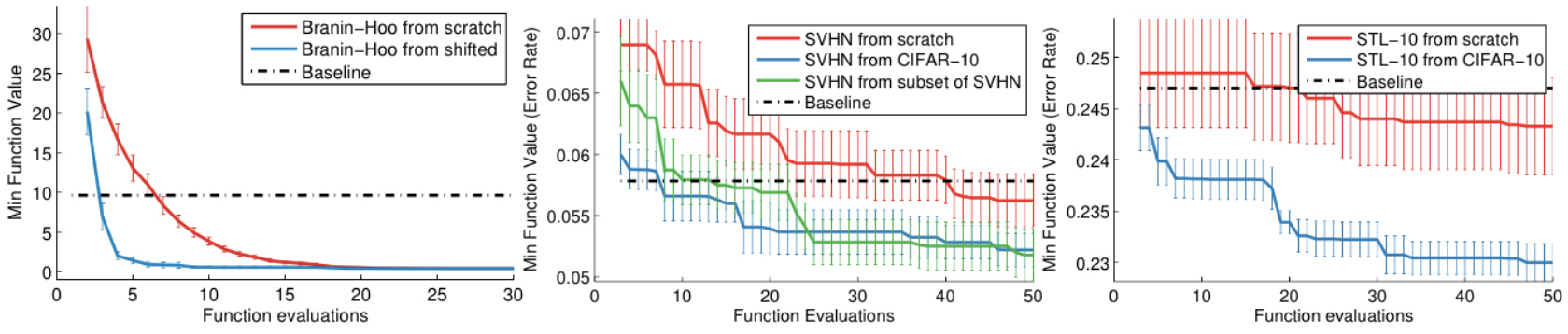
[Wersky et al., 2013]

- In other cases we want to optimize several tasks at the same time.
- We need to use a combination of them (the mean, for instance) or have a look to the Pareto frontiers of the problem.

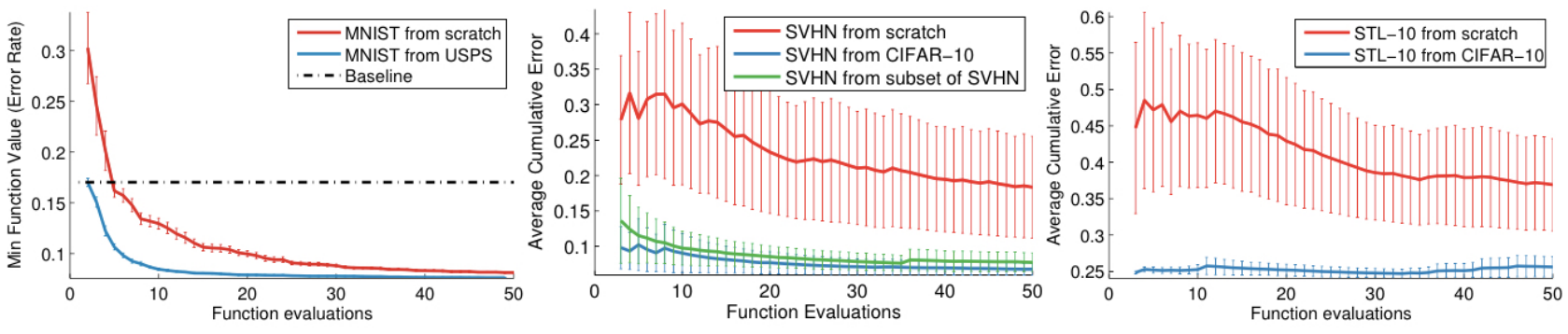
Averaged expected improvement.

# Multi-task Bayesian Optimization

[Wersky et al., 2013]




(a) Shifted Branin-Hoo (b) CNN on SVHN (c) CNN on STL-10




(d) LR on MNIST (e) SVHN ACE (f) STL-10 ACE

# Early-stopping Bayesian optimization

Swersky et al. [2014]

Considerations:

- When looking for a good parameters set for a model, in many cases each evaluation requires of a inner loop optimization.
- Learning curves have a similar (monotonically decreasing) shape.
- Fit a meta-model to the learning curves to predict the expected performance of sets of parameters

Main benefit: allows for early-stopping

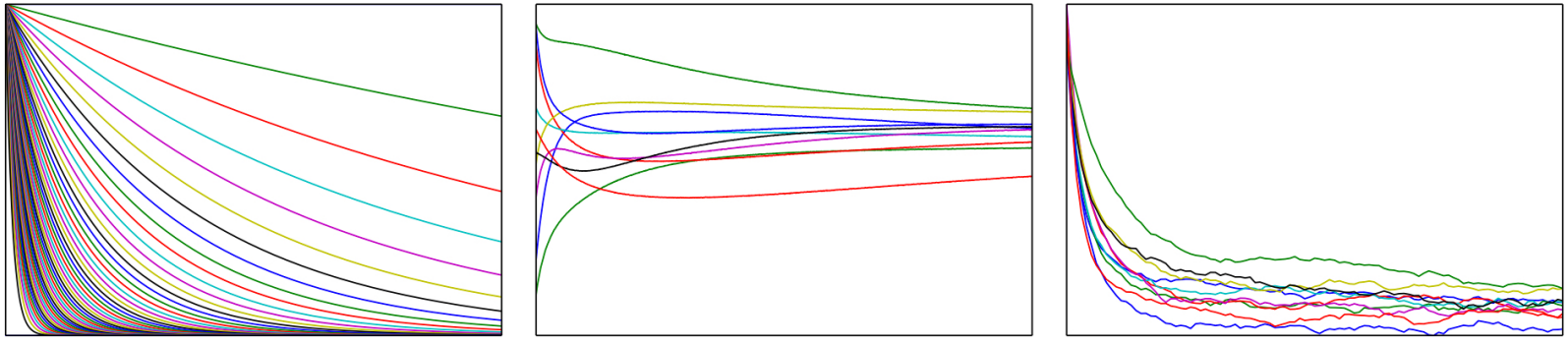
# Early-stopping Bayesian optimization

Swersky et al. [2014]

Kernel for learning curves

$$k(t, t') = \int_0^\infty e^{-\lambda t} e^{-\lambda t} \varphi(d\lambda)$$

where $\varphi$ is a Gamma distribution.

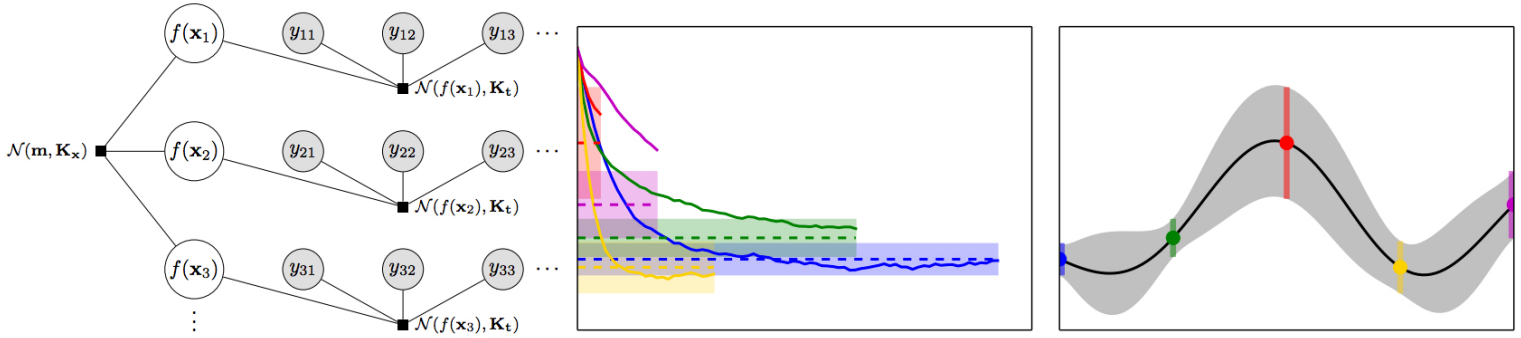


(a) Exponential Decay Basis (b) Samples (c) Training Curve Samples

# Early-stopping Bayesian optimization

Swersky et al. [2014]

- Non-stationary kernel as an infinite mixture of exponentially decaying basis function.
- A hierarchical model is used to model the learning curves.
- Early-stopping is possible for bad parameter sets.

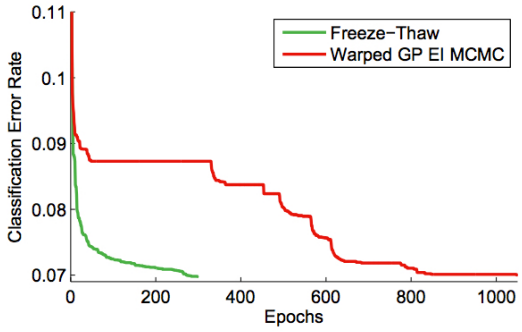



(a) Graphical Model (b) Training curve predictions (c) Asymptotic GP

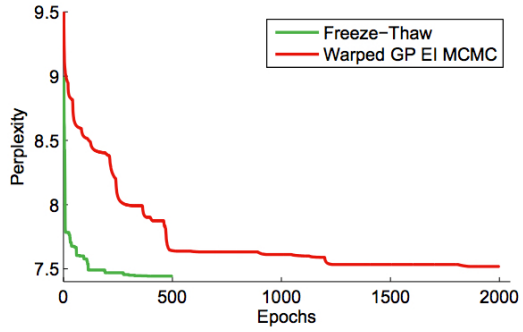
# Early-stopping Bayesian optimization

Swersky et al. [2014]

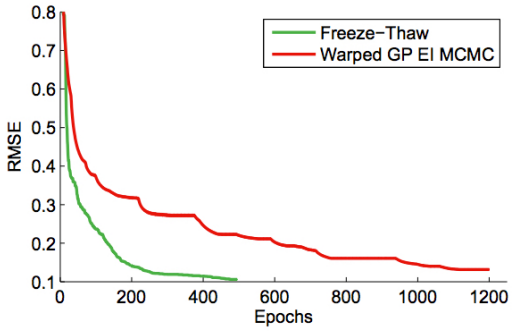
- Good results compared to standard approaches.
- What to do if exponential decay assumption does not hold?




(a) Logistic Regression




(b) Online LDA




(c) PMF

# 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.

# Considerations when designing a batch

- Available pairs $\{(\mathbf{x}_j, y_i)\}_{i=1}^n$ are augmented with the evaluations of $f$ on $\mathcal{B}_t^{n_b} = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,nb}\}$.

- Goal: design $\mathcal{B}_1^{n_b}, \ldots, \mathcal{B}_m^{n_b}$.

Notation:

- $\mathcal{I}_n$: represents the available data set $\mathcal{D}_n$ and the $\mathcal{GP}$ structure when $n$ data points are available ($\mathcal{I}_{t,k}$ in the batch context).

- $\alpha(\mathbf{x}; \mathcal{I}_n)$: generic acquisition function given $\mathcal{I}_n$.

# Available approaches

[Azimi et al., 2010; Desautels et al., 2012; Chevalier et al., 2013; Contal et al. 2013]

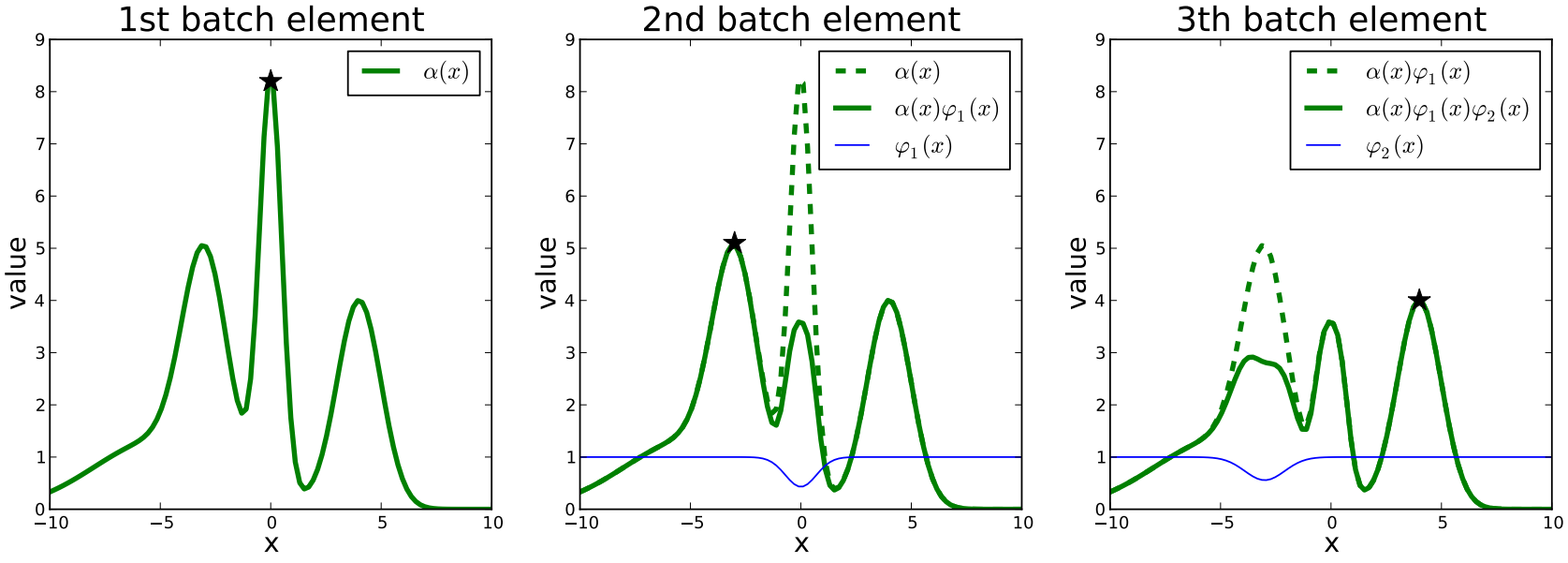
- Exploratory approaches, reduction in system uncertainty.
- Generate 'fake' observations of $f$ using $p(y_{t,j}|\mathbf{x}_j, \mathcal{I}_{t,j-1})$.
- Simultaneously optimize elements on the batch using the joint distribution of $y_{t_1}, \dots y_{t,nb}$.

**Bottleneck:** All these methods require to iteratively update $p(y_{t,j}|\mathbf{x}_j, \mathcal{I}_{t,j-1})$ to model the iteration between the elements in the batch: $\mathcal{O}(n^3)$

How to design batches reducing this cost? Local penalization

# Local penalization strategy

[González, Dai, Hennig, Lawrence, 2016]



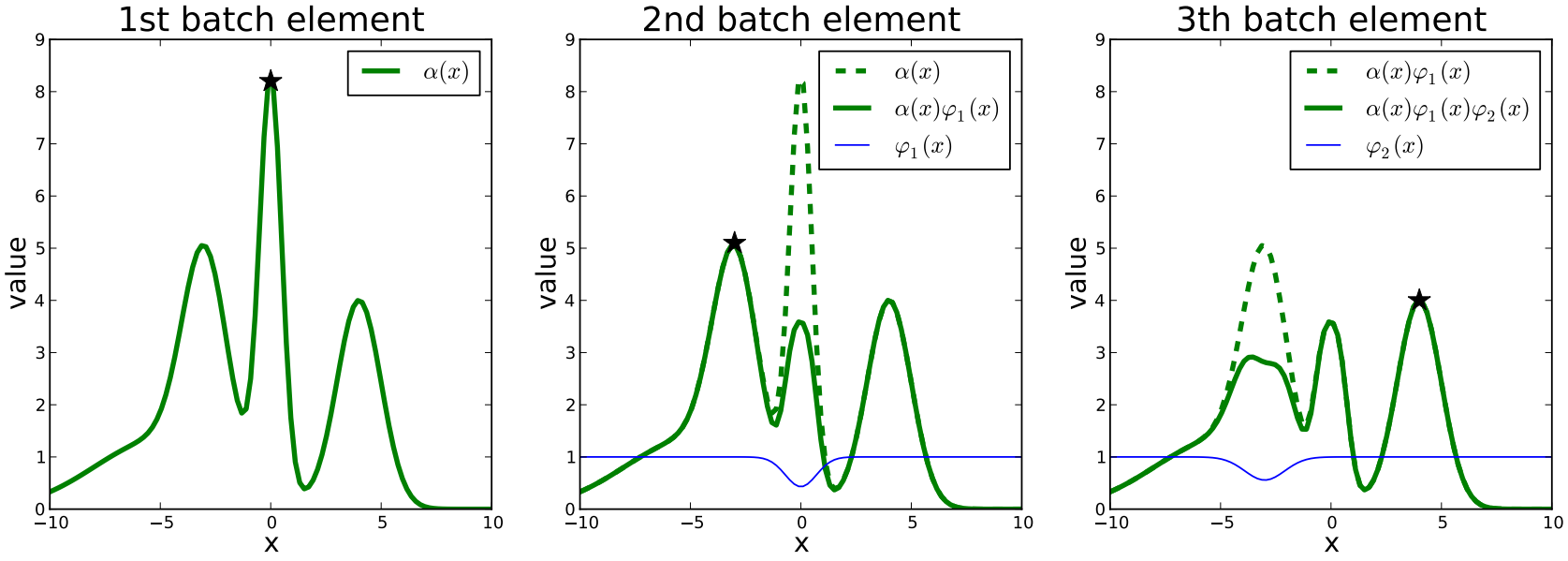

The maximization-penalization strategy selects $\mathbf{x}_{t,k}$ as

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

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

# Local penalization strategy

[González, Dai, Hennig, Lawrence, 2016]



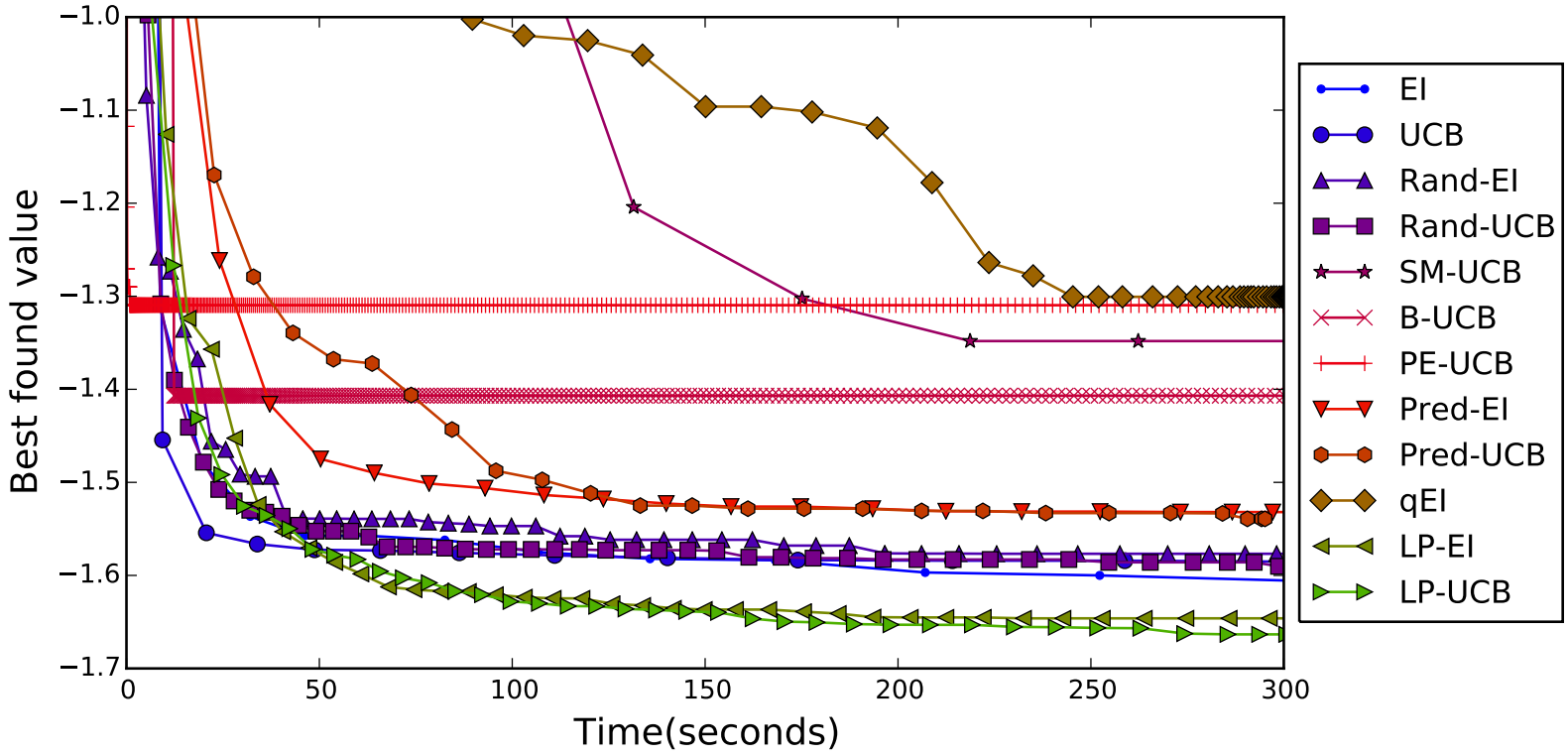

The maximization-penalization strategy selects $\mathbf{x}_{t,k}$ as

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

$g$ 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



# Myopia of optimisation techniques

[Gonzalez et al., 2016]

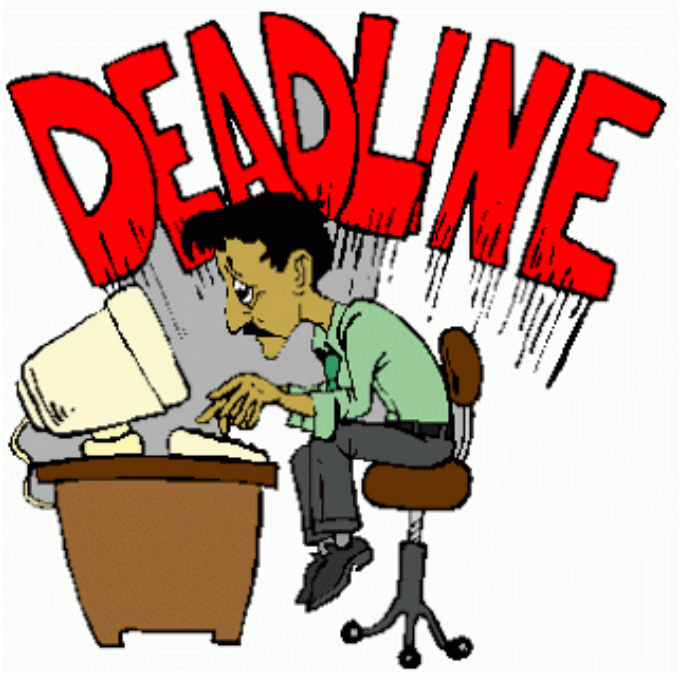
- Most global optimisation techniques are myopic, in considering no more than a single step into the future.
- Relieving this myopia requires solving the *multi-step lookahead* problem.

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

# Non-myopic thinking

[Gonzalez et al., 2016]

*To think non-myopically is important: it is a way of integrating in our decisions the information about our available (limited) resources to solve a given problem.*



# Relieving the myopia of Bayesian optimisation

[Gonzalez et al., 2016]

GLASSES!

***G**lobal optimisation with **L**ook-**A**head through **S**tochastic **S**imulation and **E**xpected-loss **S**earch*

**Idea**: jointly model the epistemic uncertainty about the steps ahead using some defining *some* point process.

# Results in a benchmark of objectives

[Gonzalez et al., 2016]

| | 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 |

GLASSES is overall the best method.

# Structured input space

[Jenatton et al., 2017]

Conditional relationships: $\mathcal{X} = \mathcal{X}_0 \times \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$

**Definition**: Depending on some values in $\mathcal{X}_i$, parameters in $\mathcal{X}_j$ become irrelevant

**Examples**:

- Feedforward neural nets:

$$\mathcal{X} = \underbrace{\mathcal{X}_0}_{\text{\# hidden layers}} \times \overbrace{\mathcal{X}_1}^{\text{hyperpar. for layer 1}} \times \underbrace{\mathcal{X}_2}_{\text{hyperpar.for layer 2}} \times \cdots \times \mathcal{X}_d$$

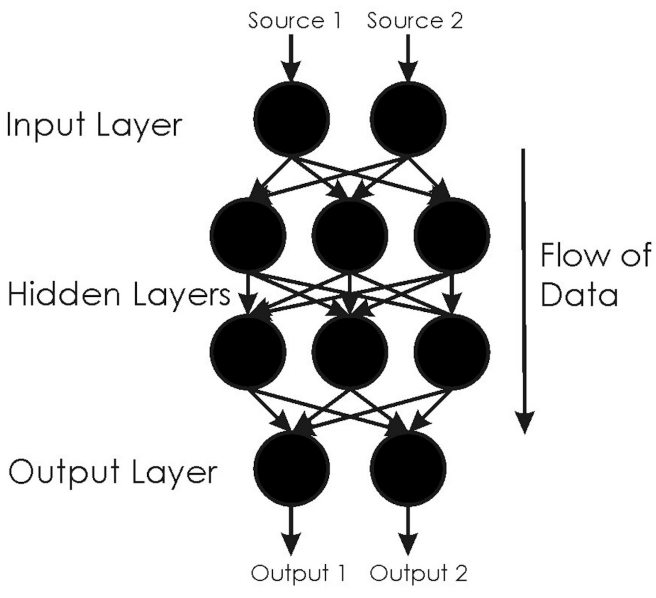
- Data analytic pipeline:

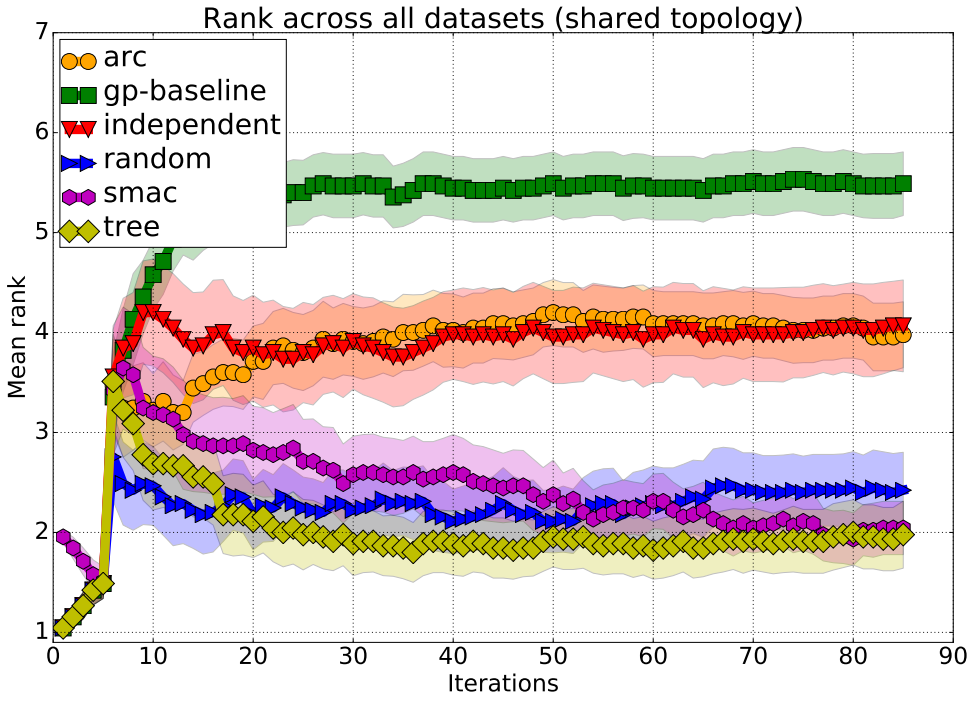
$$\mathcal{X} = \underbrace{\mathcal{X}_0}_{\text{classifier choice}} \times \overbrace{\mathcal{X}_1}^{\text{logistic-reg. hyperpar.}} \times \underbrace{\mathcal{X}_2}_{\text{random-forests hyperpar.}} \times \cdots \times \mathcal{X}_d$$

# Tree GPs based approach

[Jenatton et al., 2017]

Optimization the topology of a multilayer perceptron for classification over 45 datasets.



# Preferential Bayesian optimization

Javier González et al., 2017

- In some experiments we only have access to preferential outputs.
- These can be modeled by a GP an optimize a latent preference function.
- State-of-the-art method for learning preferences.

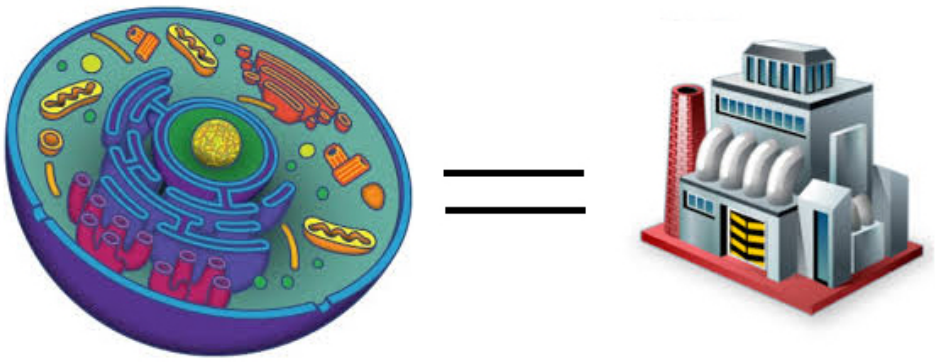
# A couple of applications

- Robotics
- Gene design

# Robotics Video

# Optimizing gene designs for drug production

[González et, 2015]



- 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.

# Take home messages

- Bayesian optimization is a way of encoding our beliefs about a property of a function (the minimum) and sequentially making decisions to know more about it.
- Two key elements: the model and the acquisition function.
- The key is to find a good balance between exploration and exploitation of the minimum in the search.

# Want to join the BayesOpt community?

- 2017: NIPS workshop in Bayesian optimization for Science and Engineering (**Deadline: End of October 2017**).
- 2018: JMLR Special Issue in Bayesian optimization (**Deadline: 31 March 2018**).