Introduction to Bayesian Optimisation

Javier González

September 2021

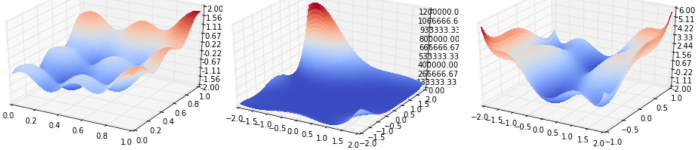
Microsoft Research Cambridge

2021 Gaussian process summer school

Problem definition

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

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

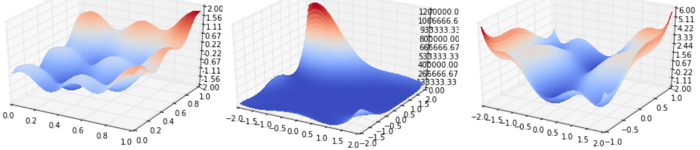


- f is explicitly unknown and multimodal.
- Evaluations of f may be perturbed by noise.
- Evaluations of f are expensive (time or cost).
- No gradient information.

Problem definition

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

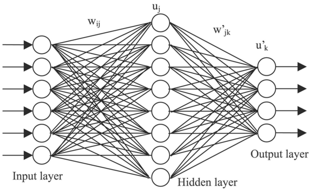
$$x_M = \arg\min_{x \in \mathcal{X}} f(x).$$



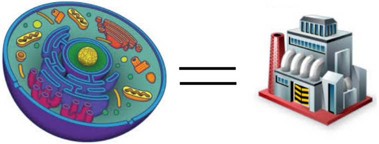
- f is explicitly unknown and multimodal.
- Evaluations of f may be perturbed by noise.
- Evaluations of f are expensive (time or cost).
- No gradient information.

Expensive functions, who doesn't have one?

Model configuration in machine learning: find optimal hyper-parameter values, learning rates, number of layers, etc.



Adaptive experimentation: Optimize a function embodied in a physical/biological process.



Expensive functions, who doesn't have one?

Many other problems:

- Robotics.
- Control, reinforcement learning.
- A/B testing.
- Scheduling, planning.
- Compilers, hardware, software.
- Industrial design.
- Intractable likelihoods.
- Simulation-optimization.

What to do to optimize a black-box function?

Option 1: Use previous knowledge

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

What to do to optimize a black-box function?

Option 2: Grid search?

f is L-Lipschitz continuous, $|f(x) - f(x')| \leq L\|x - x'\|$, 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:

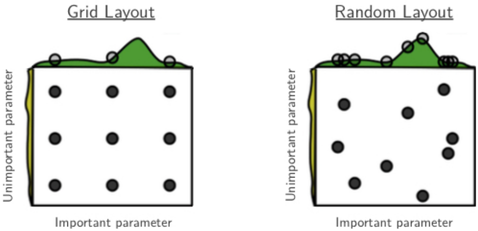
$$(L/\epsilon)^D evaluations!$$

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

What to do to optimize a black-box function?

Option 3: Random search?

We can sample the space uniformly



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

[(Image source) Bergstra and Bengio, 2012]

What to do to optimize a black-box function?

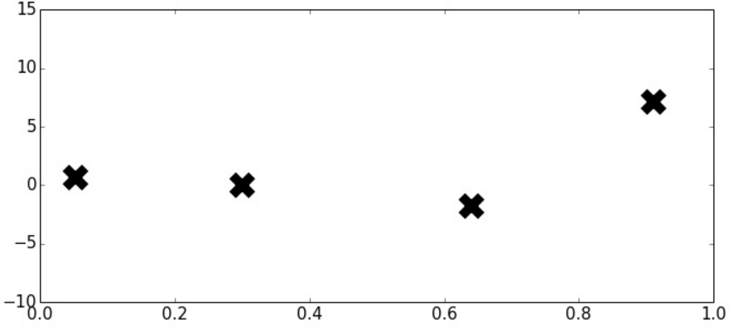
Key question:

Can we do better?

Problem (the audience is encouraged to participate!)

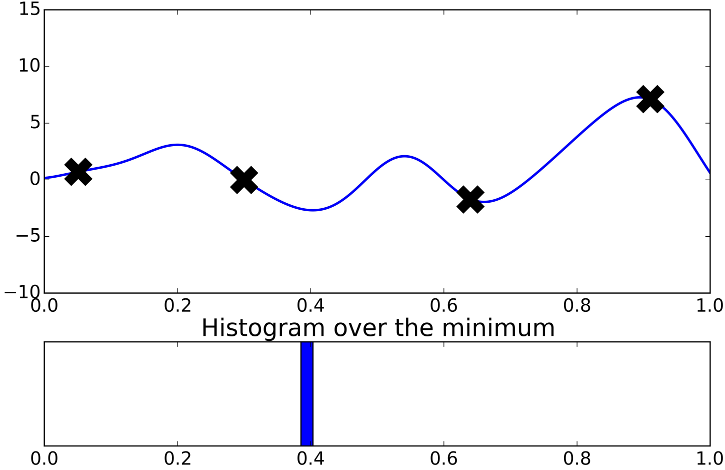
- Find the minimum of some function f in the interval [0,1].
- f is (L-Lipschitz) continuous and differentiable.
- Evaluations of f are exact and we have 4 of them!

Situation

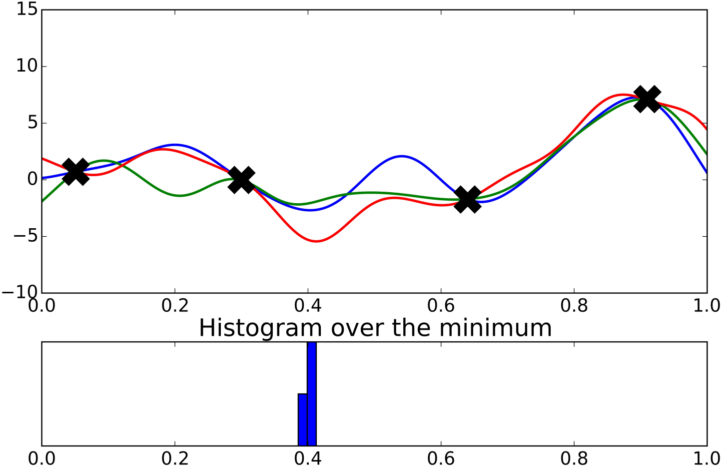


Where is the minimum of f?
Where should we take the next evaluation?

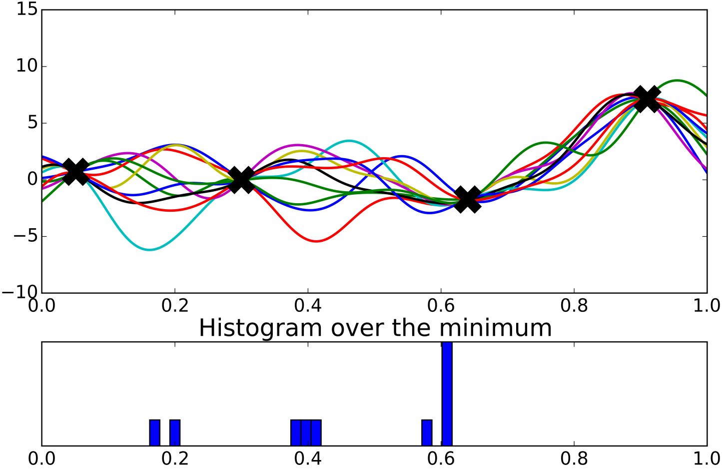
Intuitive solution



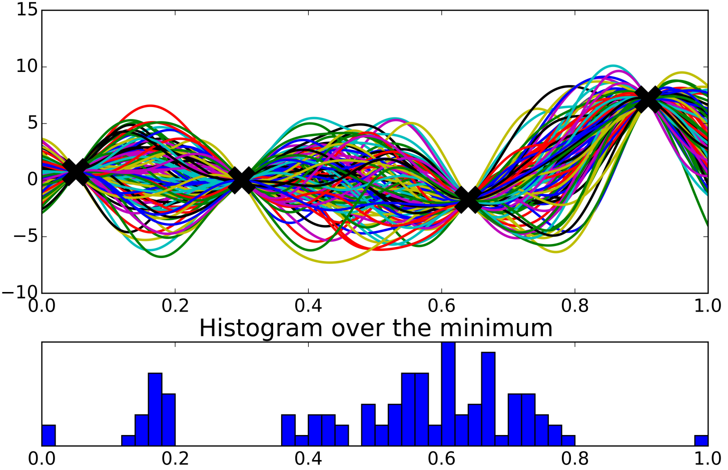
Intuitive solution



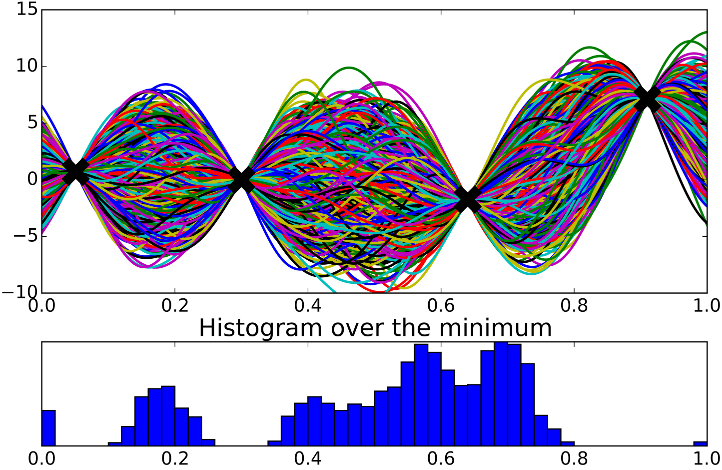
Intuitive solution



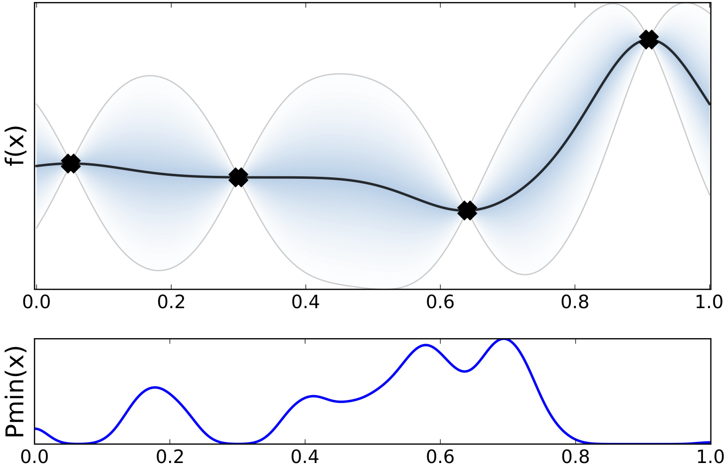
Intuitive solution



Intuitive solution



Intuitive solution



Surrogate modelling for optimization

1. Use a surrogate model of f.

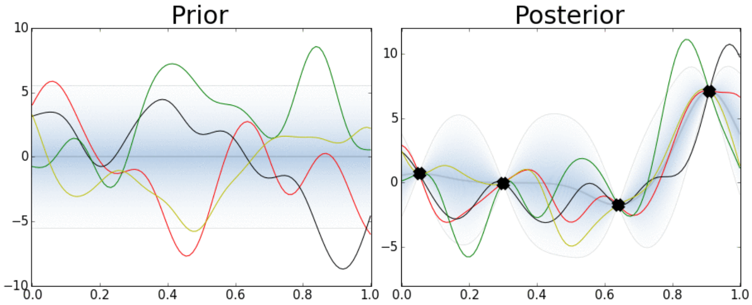
2. Define some utility/loss function to collect new data points satisfying some optimality criterion: *optimization* as *decision*.

3. Study each *decision* problems (of collecting a new point) as *inference* using the surrogate model. Calibrate both, epistemic and aleatoric uncertainty.

The surrogate model

Gaussian process emulators $f(x) \sim \mathcal{GP}(\mu(x), k_\theta(x, x'))$

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



- Model is fully determined by $\mu(x)$ and $k_\theta(x, x')$.
- Posterior can be computed in closed form.
- Uncertainty calibration.

[Rasmunsen and Williams, 2006]

Semi-mechanistic Gaussian processes

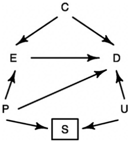
Differential equations

$$\frac{dy}{dx} = f(x)$$

$$\frac{dy}{dx} = f(x, y)$$

$$x_1 \frac{\partial y}{\partial x_1} + x_2 \frac{\partial y}{\partial x_2} = y$$

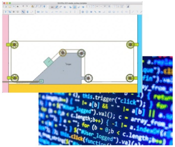
Causal graphs

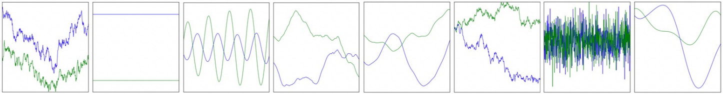


Context-free grammar

1.	E	$\rightarrow$	I
2.	E	$\rightarrow$	$E+E$
3.	E	$\rightarrow$	$E*E$
4.	E	$\rightarrow$	(E)
5.	I	$\rightarrow$	a
6.	I	$\rightarrow$	b
7.	I	$\rightarrow$	Ia
8.	I	$\rightarrow$	Ib
9.	I	$\rightarrow$	$I0$
10.	I	$\rightarrow$	$I1$

Computer code





- Model complex functions (Deep GPs are also an option).
- Kernel design: we can incorporate prior knowledge into $k_\theta(x, x')$.

Other models are also valid

- T-Student processes.
- Random Forests.
- Bayesian neural networks.
- Trees of Parzen estimators.
- etc.

Any model able to calibrate uncertainty (needed for exploration) can be used in Bayesian optimization.

Exploration vs. exploitation

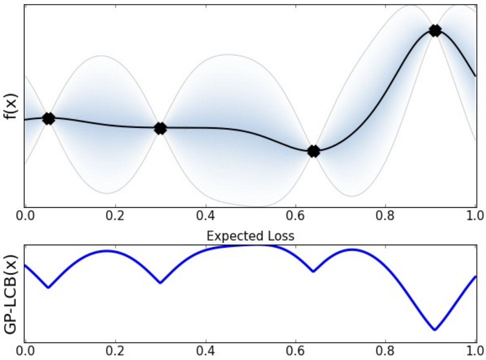
The exploration-exploitation dilemma is present in most of our day-by-day decisions.

Bayesian reasoning

The acquisition function

GP Upper (lower) Confidence Band

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

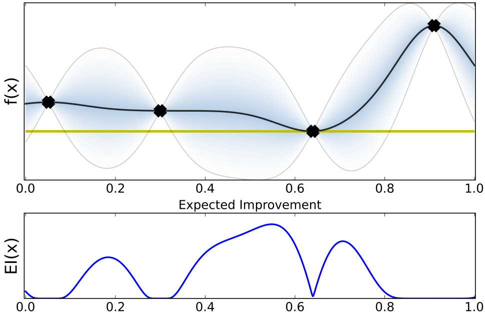


- Upper (lower) bounds f, theoretical results are available.
- Optimal choices available for the 'regularization parameter'.
- Direct balance between exploration and exploitation.

[Srinivas et al., 2010]

Expected improvement

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

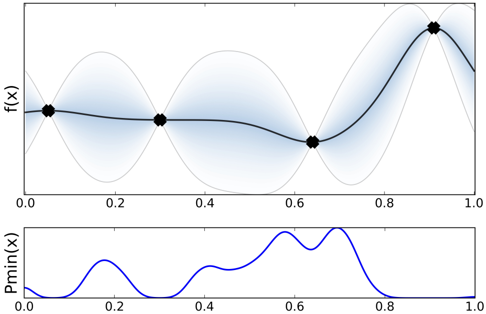


- Perhaps the most used acquisition.
- Explicit form available for Gaussian posteriors.
- It is too greedy in some problems.

[Jones et al., 1998]

Entropy search and Predictive Entropy search

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

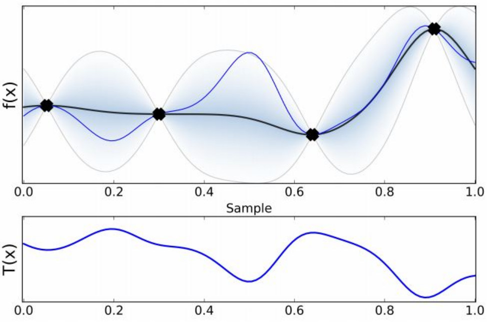


- Information theoretic approaches: reduce the entropy of $p(x_{min})$.
- Same acquisition, two different approximations (ES, PES).
- Approximating $p(x_{min})$ is not trivial.

[Hennig et al., 2013; Lobato et al., 2014]

Thompson sampling

$\alpha_{THOMP.}(\mathbf{x};\theta,\mathcal{D}) = g(\mathbf{x})$, where $g(\mathbf{x})$ is sampled form $\mathcal{GP}(\mu(x), k(x,x'))$.



- Stochastic acquisition function.
- Used in PES to compute $p(x_{min})$.
- Uses Fourier features for continuous samples.

[Rahimi and B. Recht, 2007]

Other acquisitions

Each acquisition balances exploration-exploitation in a different way. No universal best method.

Others:

- Probability of improvement.
- Knowledge gradient.
- Approximations of Max-value entropy search (MES, GIBBON).
- etc.

[Hushner, 1964; Wu et al., 2017; Wang and Jegelka, 2017; Moss et al., 2021]

The algorithm

Bayesian Optimisation

Choose a prior measure over f and collect some initial data.

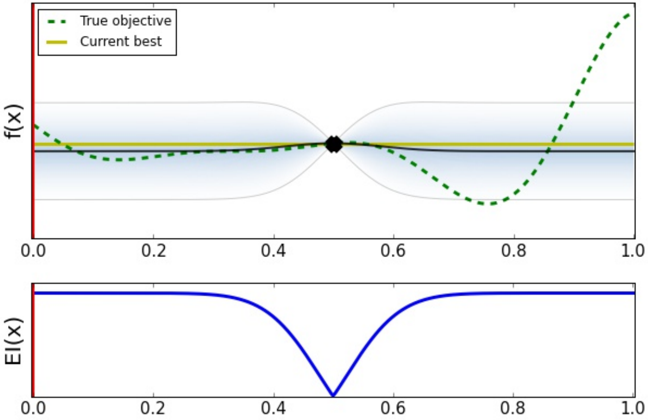
While the budget is not over:

1. Combine the prior and the available data to get a posterior.
2. Use the posterior to build a acquisition/loss function.
3. Optimize the acquistion and augment the dataset.

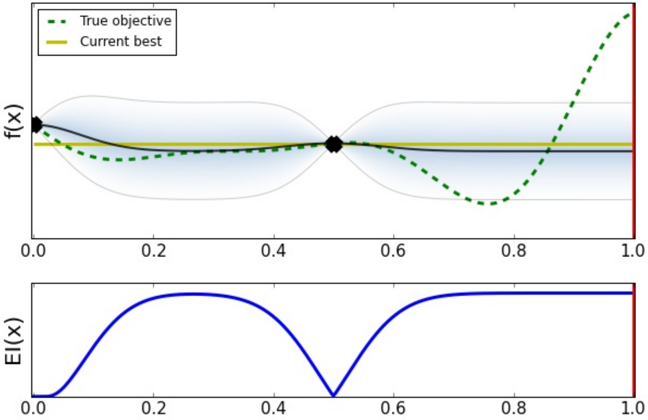
Report best found location.

[Mockus, 1978]

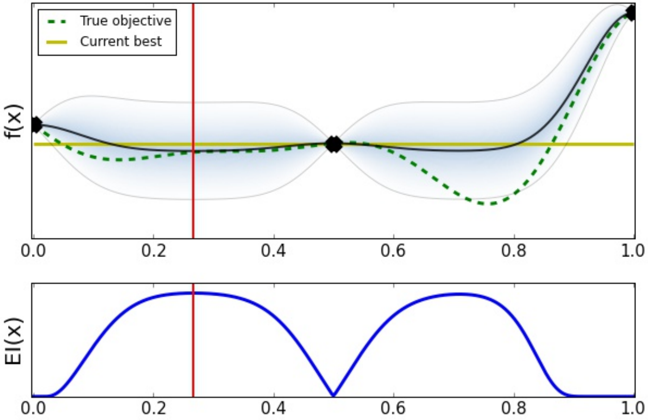
Bayesian optimization in action



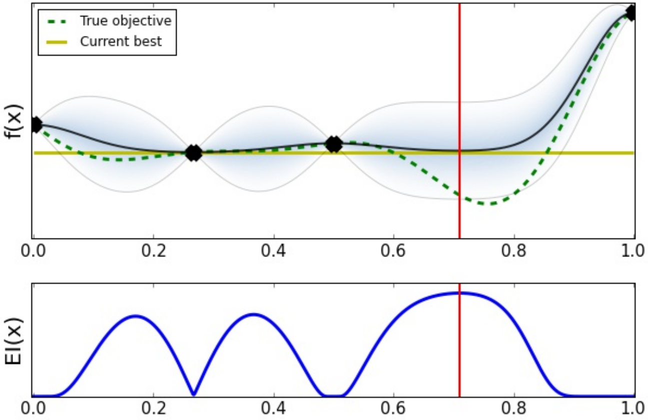
Bayesian optimization in action



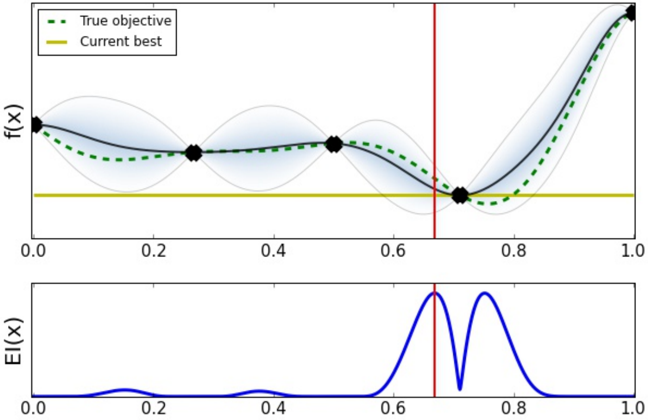
Bayesian optimization in action



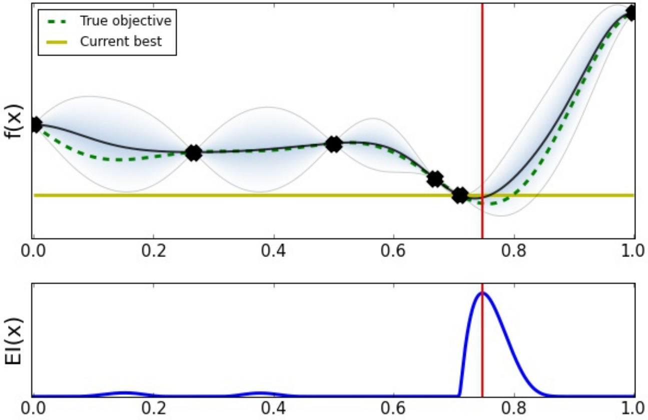
Bayesian optimization in action



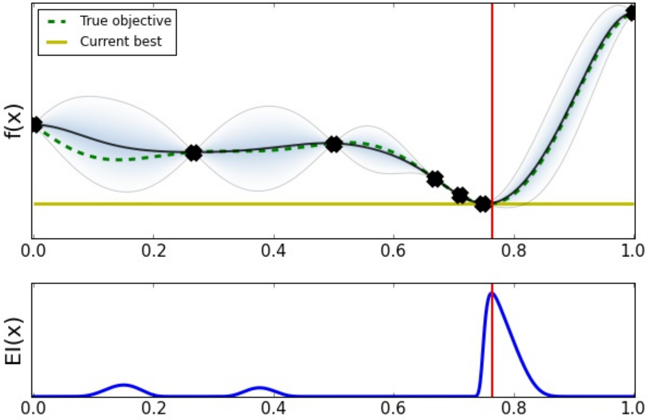
Bayesian optimization in action



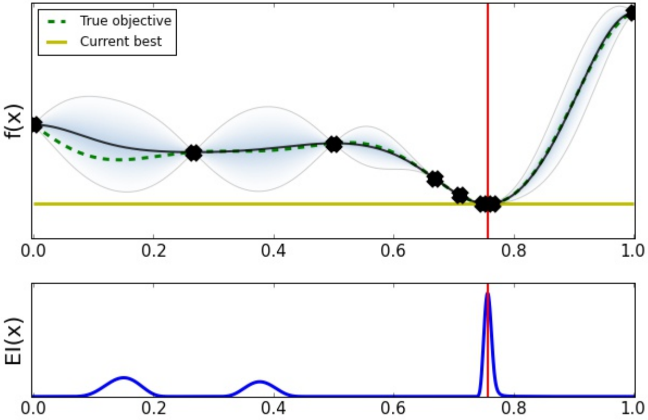
Bayesian optimization in action



Bayesian optimization in action



Bayesian optimization in action



Bayesian Optimization

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.
- Issure: still need to find x_{n+1} in each iteration.

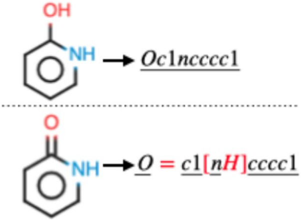
Practical considerations

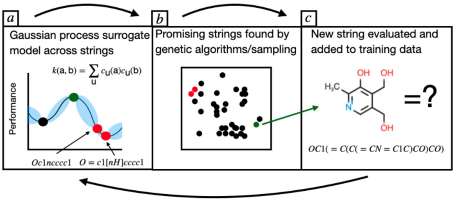
- Handle the hyper-parameters of the surrogate model.
- Picking the right covariance/model.
- Initial designs, how to start?
- Optimizing the acquisition function.

Review paper by Shahriari, et al. (2016): Taking the Human Out of the Loop: A Review of Bayesian Optimization. Proceedings of the IEEE 104(1):148–175.

Optimizing over non-Euclidean spaces

Optimizing over string spaces





- Standard BO methods are defined on Euclidean spaces.
- Optimizing over strings or other structured spaces is not trivial.
- In many relevant problems (drug design, gene optimization, etc.) the input space is defined over strings.

GPs with a string kernel

BOSS: Bayesian Optimization for String Spaces.

1. Use a GP with a string kernel:

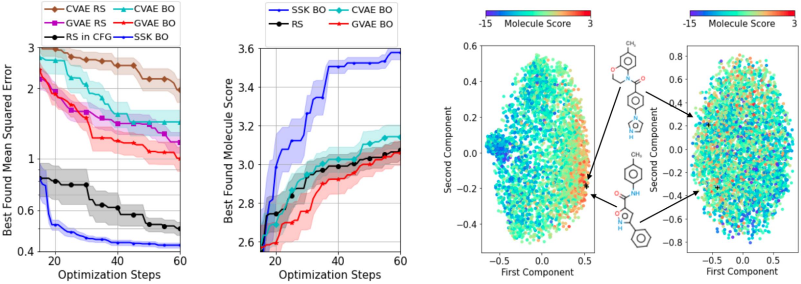
$$k_n(\mathbf{a}, \mathbf{b}) = \sum_{\mathbf{u} \in \Sigma^n} c_{\mathbf{u}}(\mathbf{a}) c_{\mathbf{u}}(\mathbf{b})$$

   - $c_{\mathbf{u}}(\mathbf{s}) = \lambda_m^{|\mathbf{u}|} \sum_{1<i_1<..<i_{|\mathbf{u}|}<|\mathbf{s}|} \lambda_g^{i_{|\mathbf{u}|}-i_1} \mathbb{I}_{\mathbf{u}}((s_{i_1}, .., s_{i_{|\mathbf{u}|}}))$.
   - Σ^n set of all possible ordered collections alphabet Σ.
   - $\mathbb{I}_{\mathbf{x}}(\mathbf{y})$ indicator function checking if the strings $\mathbf{x}$ and $\mathbf{y}$ match.
   - Match decay $\lambda_m \in [0, 1]$ and gap decay $\lambda_g \in [0, 1]$.

2. Optimize the acquisition function with a genetic algorithm:
   - Unconstrained spaces and locally constrain spaces.
   - Grammar constrain spaces.
   - Candidate set.

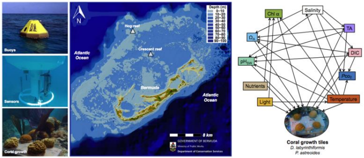
[Moss et al, 2020]

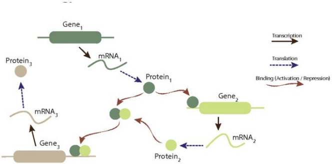
Results

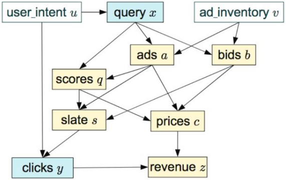


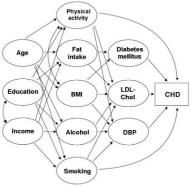
- State-of-the-art approach compared to other alternatives (VAEs, feature based representations, etc.).
- Only two parameters to tune in the model.

Optimizing the output of a causal graph



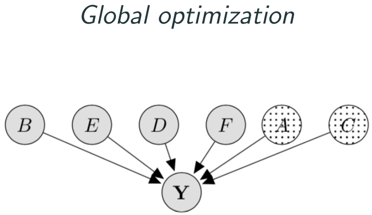


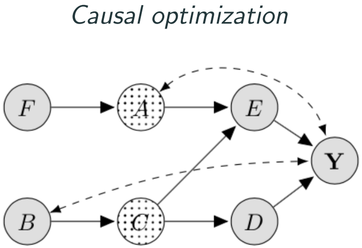




[González, 2015; Maksimov, 2015; Murray et al, 2003; Courtney et al, 2017; Bottou et al, 2013]

Global optimization vs. Causal optimization





Idea: Use the topology of the graph to find the minimal subsets of variables that need to be tuned to optimize the output Y.

[Aglietti et al, 2020]

Causal Bayesian optimization

Explore vs. exploit; observe vs. intervene.

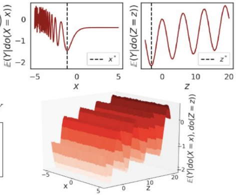
Algorithm 1: Causal Bayesian Optimization-CBO

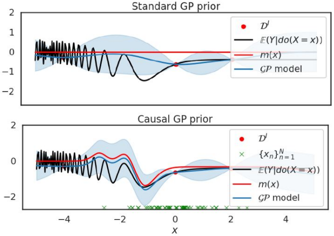
Data: $\mathcal{D}^{\mathrm{O}}$, $\mathcal{D}^{\mathrm{I}}$, $\mathcal{G}$, **ES**, number of steps T
Result: $\mathbf{X}_s^\star, \mathbf{x}_s^\star, \hat{\mathbb{E}}[\mathbf{Y}^\star|\mathrm{do}(\mathbf{X}_s^\star = \mathbf{x}_s^\star), \mathbf{C}]$
Initialise: Set $\mathcal{D}_0^{\mathrm{I}} = \mathcal{D}^{\mathrm{I}}$ and $\mathcal{D}_0^{\mathrm{O}} = \mathcal{D}^{\mathrm{O}}$
for t=1, ..., T **do**
  Compute ϵ and sample $u \sim \mathcal{U}(0, 1)$
  if $\epsilon > u$ **then**
    (Observe)
    1. Observe new observations $(\mathbf{x}_t, c_t, \mathbf{y}_t)$.
    2. Augment $\mathcal{D}^{\mathrm{O}} = \mathcal{D}^{\mathrm{O}} \cup \{(\mathbf{x}_t, c_t, \mathbf{y}_t,)\}$.
    3. Update prior of the causal GP (Eq. (2)).
  end
  else
    (Intervene)
    1. Compute $EI^s(\mathbf{x})/Co(\mathbf{x})$ for each element $s \in$ **ES** (Eq. (5)).
    2. Obtain the optimal interventional set-value pair $(s^\star, \alpha^\star)$.
    3. Intervene on the system.
    4. Update posterior of the interventional GP.
  end
end
Return the optimal value $\hat{\mathbb{E}}[\mathbf{Y}^\star|\mathrm{do}(\mathbf{X}_s^\star = \mathbf{x}_s^\star), \mathbf{C}]$ in $\mathcal{D}_T^{\mathrm{I}}$ and the corresponding $\mathbf{X}_s^\star, \mathbf{x}_s^\star$.

$X \to Z \to Y$

$$X = \epsilon_X$$
$$Z = \exp(-X) + \epsilon_Z$$
$$Y = \cos(Z) - \exp(-\frac{Z}{20}) + \epsilon_Y$$

$$\boxed{\begin{aligned} \mathbb{M}_{\mathcal{G},Y} &= \{\varnothing, \{X\}, \{Z\}\} \\ \mathbb{P}_{\mathcal{G},Y} &= \{\{Z\}\} \\ \mathbb{B}_{\mathcal{G},Y} &= \{\{X, Z\}\} \end{aligned}}$$

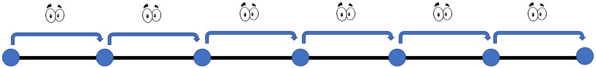




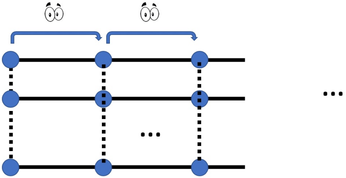
Batch Bayesian optimization

Batch Bayesian optimization

Standard Bayesian optimization



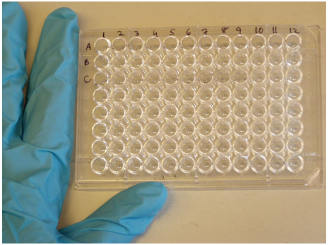
Batch Bayesian optimization



Batch Bayesian optimization

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

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



Examples: multiple cores to optimize a computer code, well plates in lab experimentation, etc.

Approaches

- Non-greedy, joint optimization of the batch $\mathcal{B}_t^{n_b}$:

$$\alpha_{qEI}(\mathbf{X};\theta,\mathcal{D}) = \int_y \max(0, y_{best} - Y) p(Y|\mathbf{X};\theta,\mathcal{D}) dY$$

  Each batch requires solving a $D \times n_b$ optimization (bad scalability).

- Greedy, sequential optimization of the batch $\mathcal{B}_t^{n_b}$:
  1. Optimize $\alpha_{EI}(\mathbf{x};\theta,\mathcal{D})$.
  2. Fantasize a value of y in that location.
  3. Find next point for the update the model.

  Number of samples scales exponentially with the size of the batch.

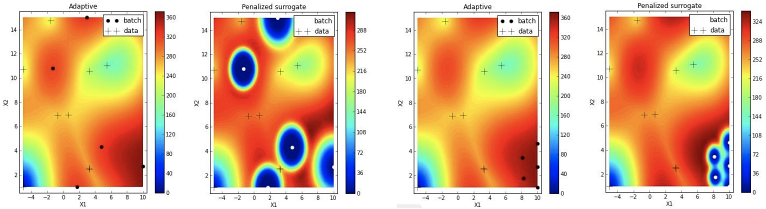
[Azimi et al., 2010; Desautels et al., 2012; Chevalier et al., 2013; Contal et al. 2013, etc.]

Local penalization strategy

The maximization-penalization strategy selects $x_{t,k}$ as

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

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

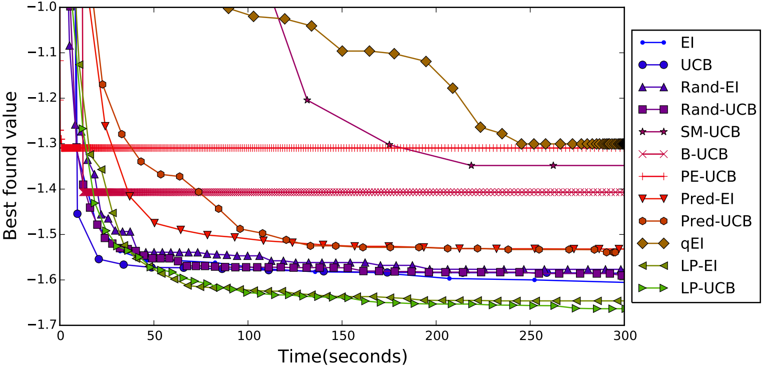


Batch of size 5 for two different values of the Lipschitz constant L

[Gonzalez et al. 2016]

2D experiment with ‘large domain’

Comparison in terms of the wall-clock time.

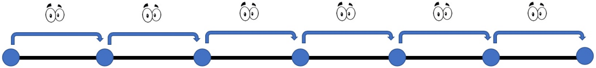


[Gonzalez et al. 2016]

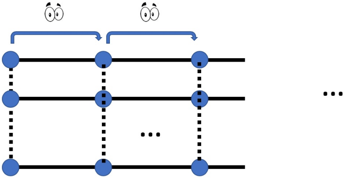
Non-myopic Bayesian optimization

Non-myopic Bayesian optimization

Standard Bayesian optimization

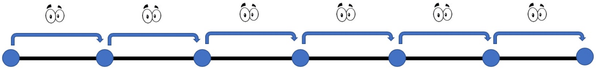


Batch Bayesian optimization

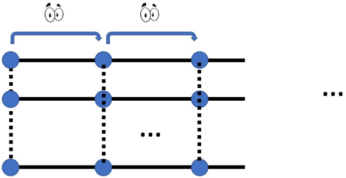


Non-myopic Bayesian optimization

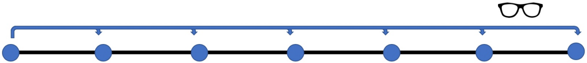
Standard Bayesian optimization



Batch Bayesian optimization

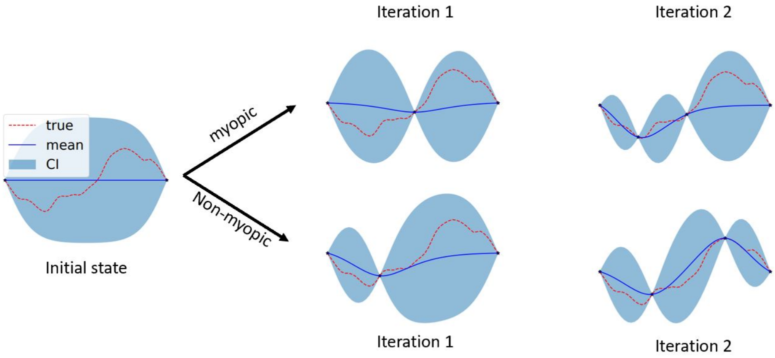


Bayesian optimization with look-ahead (non-myopic)



Illustration

Reasoning myopically is sub-optimal when we know the remaining budget.



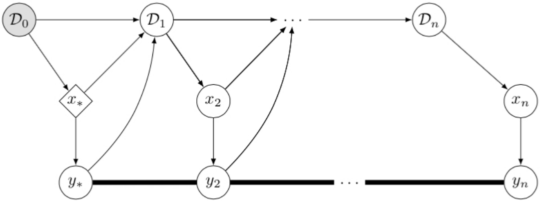
Core problem

- One-step marginal utility $\alpha(x|\mathcal{D})$:

$$v_1(x|\mathcal{D}) = \alpha(x|\mathcal{D})$$

- Multiple steps utility be decomposed with the Bellman recursion:

$$v_t(x|\mathcal{D}) = v_1(x|\mathcal{D}) + \mathbb{E}_y[\max_{x'} v_{t-1}(x'|\mathcal{D} \cup \{(x, y)\})]$$



Optimizing the non-myopic policy is intractable.

Approximations to the optimal policy

- Two-steps look-ahead:

$$v_2(x|\mathcal{D}) = v_1(x|\mathcal{D}) + \mathbb{E}_y[\max_{x'} v_1(x'|\mathcal{D}_1)]$$

- GLASSES (Global optimisation with Look-Ahead through Stochastic Simulation and Expected-loss Search):

$$v_t(x|\mathcal{D}) = v_1(x|\mathcal{D}) + \mathbb{E}_y[V_1^{t-1}(X'|\mathcal{D}_1)]$$

- BINOCULARS (Batch-Informed Non-myopic Choices, Using Long-horizons for Adaptive, Rapid SED):

$$v_t(x|\mathcal{D}) = v_1(x|\mathcal{D}) + \max_{X} \mathbb{E}_y[V_1^{t-1}(X'|\mathcal{D}_1)]$$

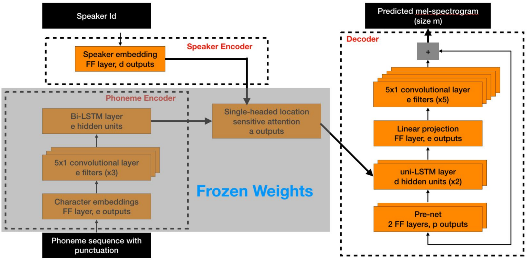
where V_t is a batch value function and X' a pre-computed batch.

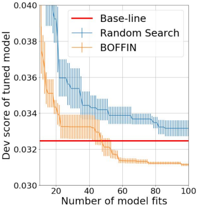
[Gonzalez et al. 2016; Jiang et al. 2019]

Applications

Learning voices with a few utterances

Fine-tune a pre-trained test-to-speech model to mimic a new speaker using a small corpus of target utterances.





(a) INTERNAL speaker A.

Full voice reconstruction with are few sentences.

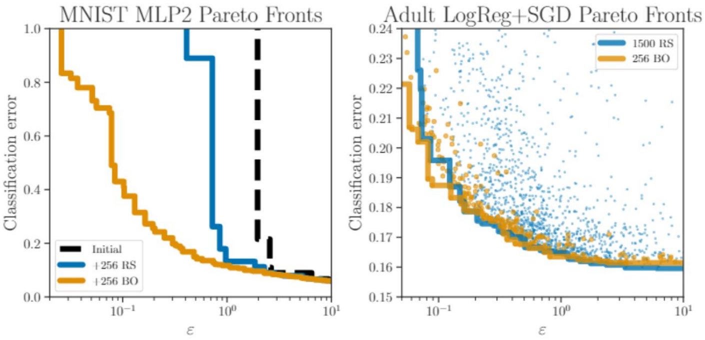
[Moss et al. 2019]

Drone design and control

Drone controller

Privacy accuracy trade-off

Optimizing the hyper-parameters of machine learning models to balance the privacy-accuracy trade off (learn the optimal Pareto front).

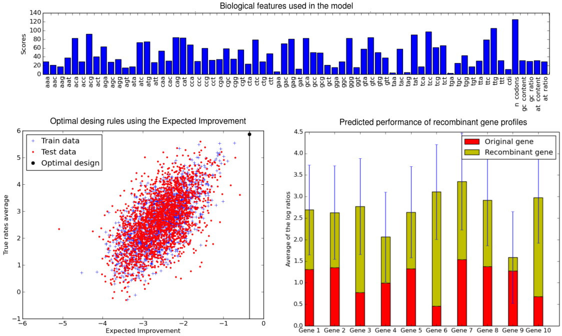


Select the best accuracy given a level of differential privacy (ϵ).

[Avent et al. 2019]

Synthetic gene design

- 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.

[Gonzalez et al, 2015]

Summary

- Simple algorithm, multiple applications.
- Two basic elements: model and acquisition.
- Proper exploration-exploitation is the key to solve real problems.
- Use domain knowledge the is key to address real problems.
- Wide range of code bases available with multiple implementations.

Questions?