Bayesian Optimization: Basics & Challenges

Zhenwen Dai

Spotify

2020-01-22

Bayesian Optimization

[Močkus, 1975]

Bayesian Optimization (BO) is a family of global optimization methods for an unknown objective function with a bounded space:

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}).$$

- The cost of obtaining a value of f at a chosen location is very expensive.
- The gradient of f is usually not available.

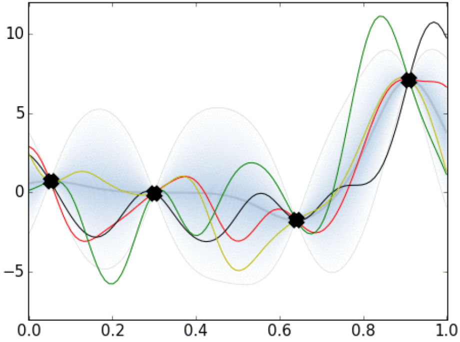
Real-world Applications

- Automated Machine Learning
- drug discovery
- design optimization

Surrogate modelling

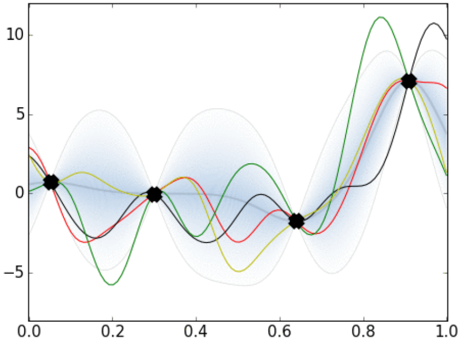
The core idea: building a probabilistic model of the unknown objective.

- Predict the value of the objective function at unseen locations *with uncertainty*.
- Explicitly encode the domain knowledge about the objective function.



Surrogate modelling: Common choices

- Gaussian process (GP) is the most common choice of surrogate model for BO.
- GP is a distribution of functions.
  `https://www.youtube.com/watch?v=VsW-eTsqBCk`
- Other probabilistic models such as random forests and bayesian neural networks have been studied as well.



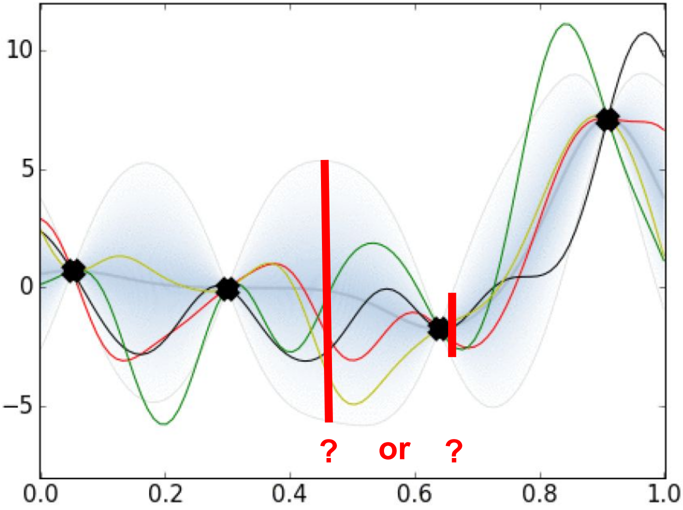
A common BO loop

1. Select a *prior* distribution about the unknown function f.
2. Estimate the *posterior* distribution of f based on the data collected so far.
3. Use the *posterior* to decide where to collect the next datapoint according to some *acquisition*/*loss* function.
4. Collect the output of f at the chosen location and augment the data.
5. Repeat from Step 2.

Run the algorithm until the budget is over.

How to choose the next location for evaluation?

Exploration vs. Exploitation



Choose the next location

The policy of BO is usually phrased as

- The value of every location in the search space is scored by a utility function based on the prediction of the surrogate model, often called acquisition function, $\alpha(\mathbf{x}; D)$.
- The next evaluation is chosen as the location having the highest value:

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}; D_t).$$

Acquisition function

Heuristic function for exploration and exploitation trade-off

- **Upper confidence bound (UCB)**

$$\alpha_{\mathsf{LCB}}(x; D) = -\mu(x; D) + \beta\sigma(x; D)$$

- **Expected improvement (EI)**

$$\alpha_{\mathsf{EI}}(x; D) = \int_y \max(0, \hat{y} - y) p(y|x, D) \mathrm{d}y$$

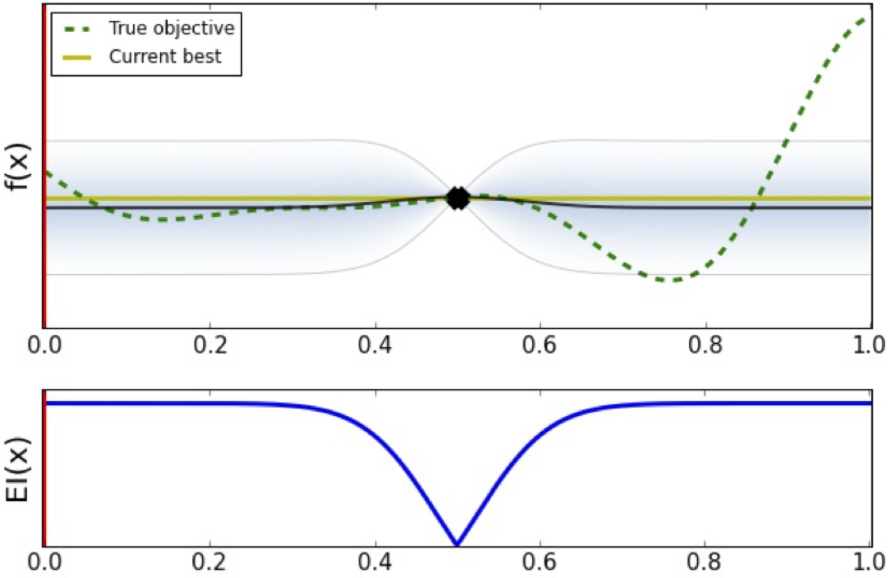
- **Thompson sampling**

$$\alpha_{\mathsf{thompson}}(x; D) = -g(x), \quad g(\cdot) \sim \mathcal{GP}(f(\cdot)|D)$$

- **Entropy search (ES)**

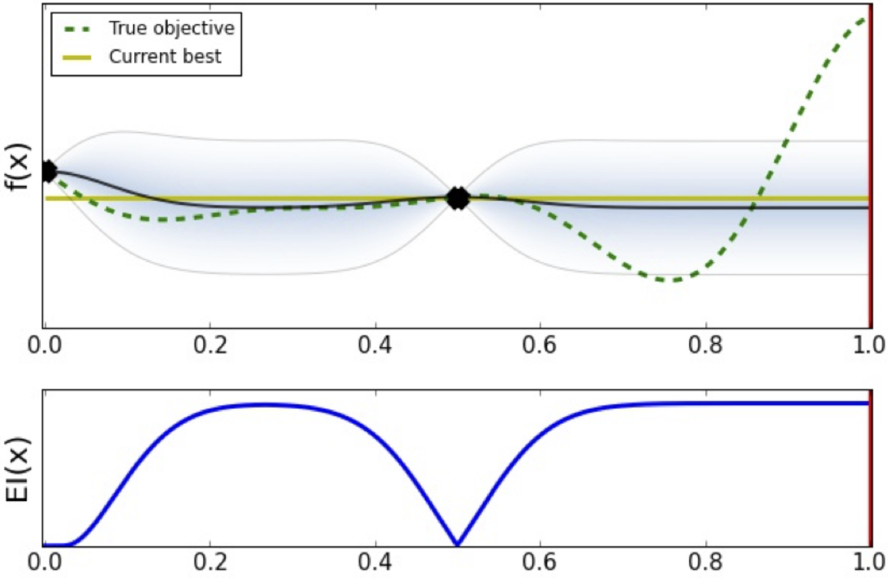
$$\alpha_{\mathsf{ES}}(x; D) = H[p(x_{\mathsf{min}}|D)] - \mathbb{E}_{p(y|D,x)}[H[p(x_{\mathsf{min}}|D \cup \{x, y\})]]$$

A BO Example

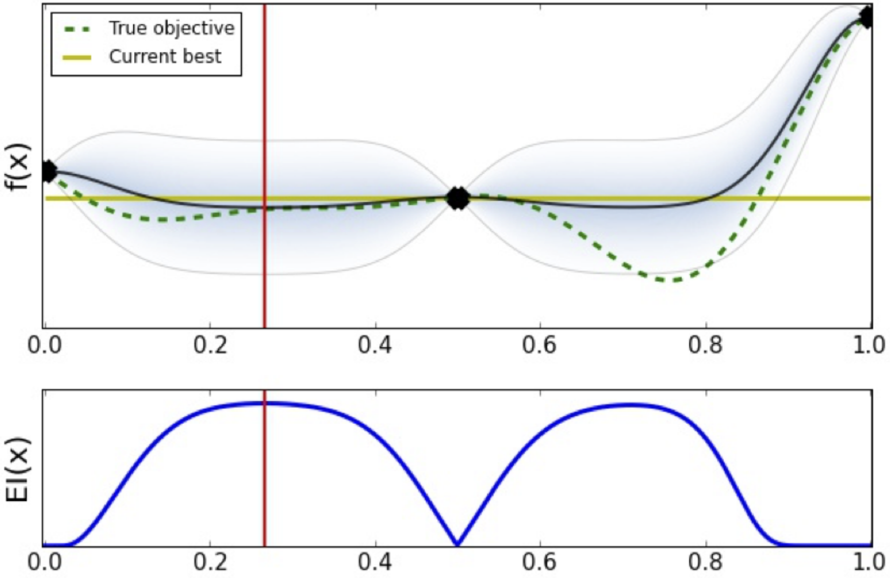


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

A BO Example

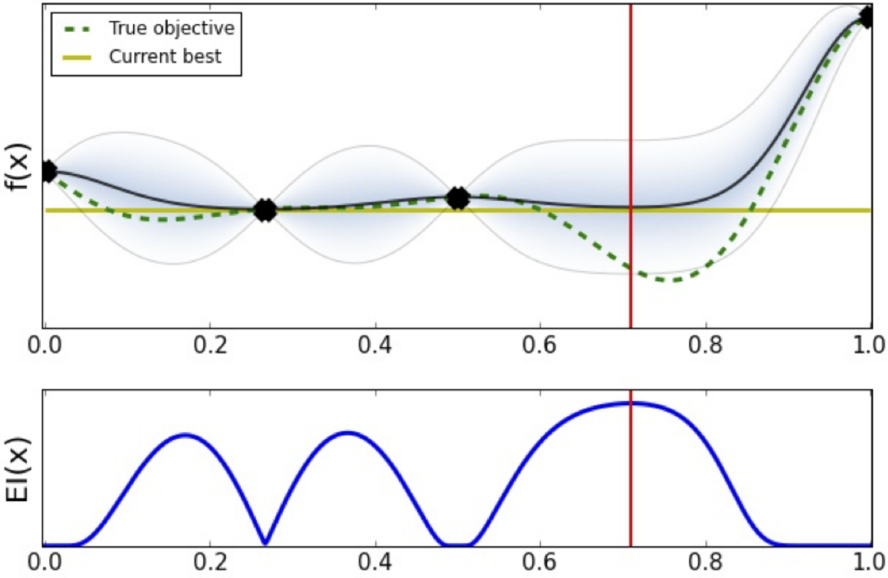

True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

A BO Example



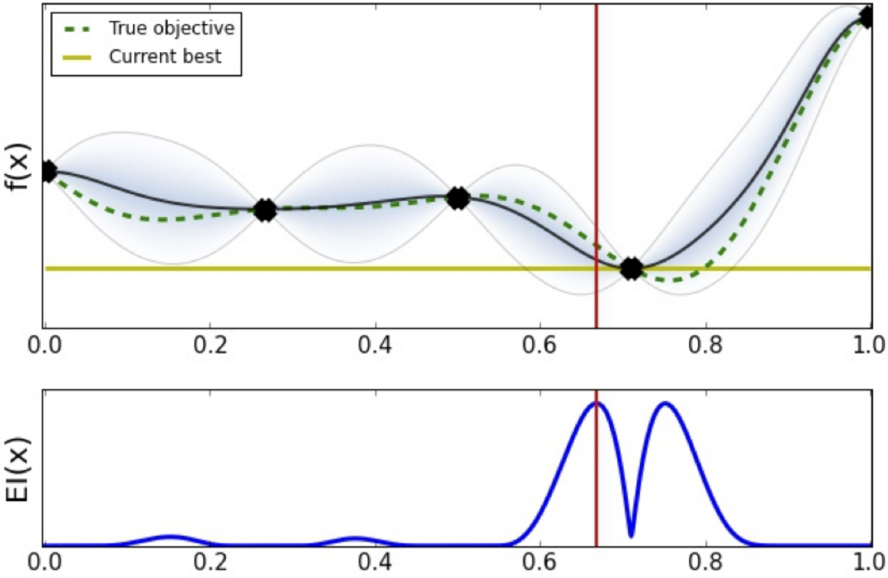
True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

A BO Example



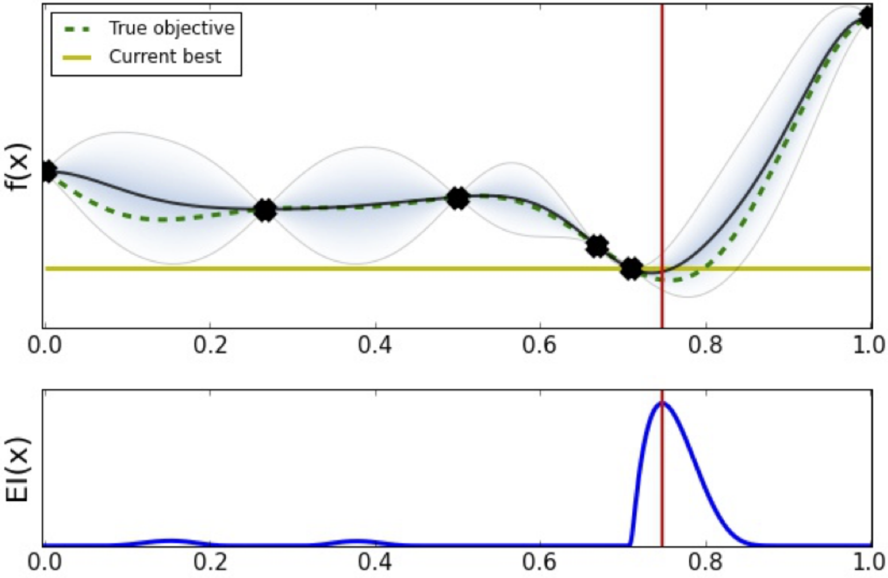
True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

A BO Example



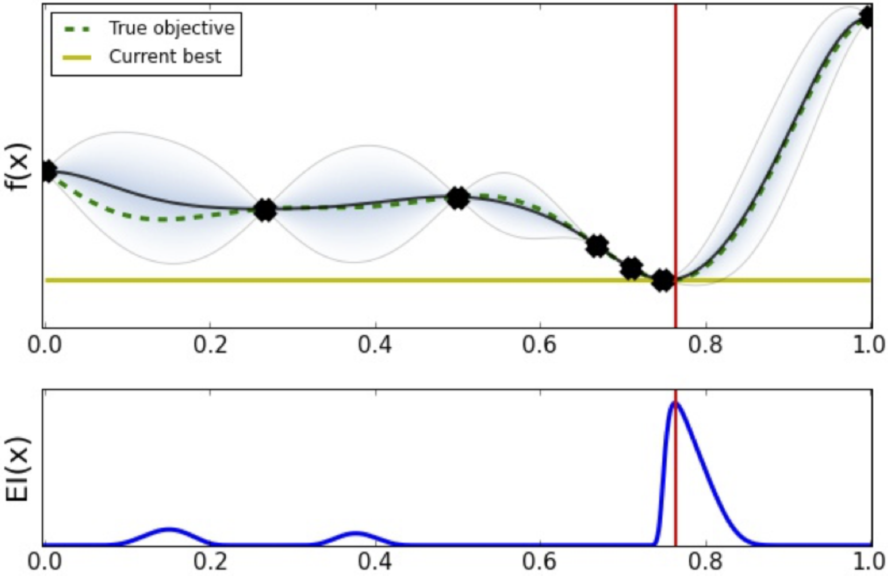
True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

A BO Example



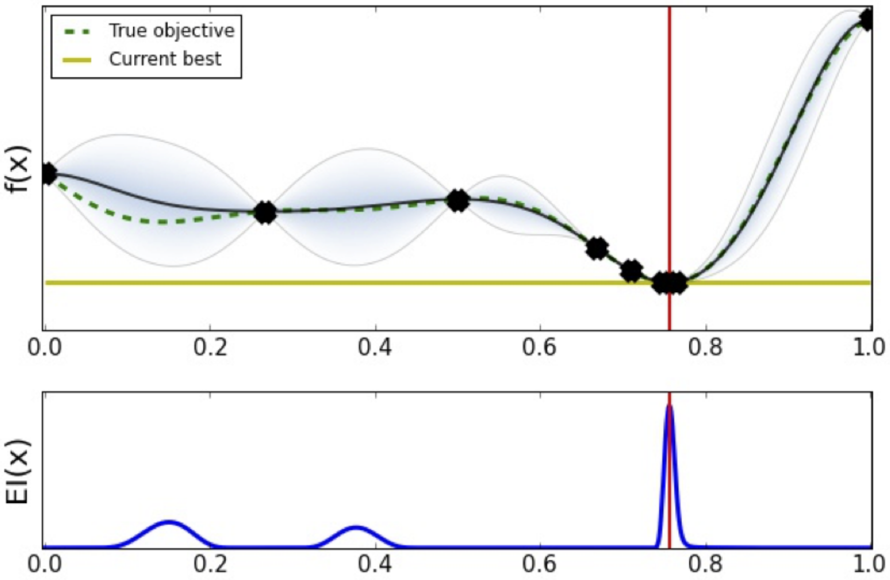
True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

A BO Example



True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

A BO Example



True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

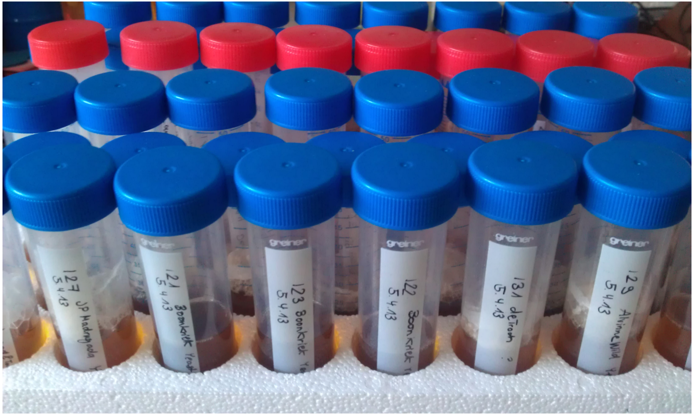
Challenges

Bayesian Optimization has shown good performance for low-dimensional (up to 20) and smooth objective functions.

Challenges:

- Batch / Parallel Evaluation
- Non-myopic
- The dimensionality of space
- Structured search space
- Multi-task/objective search, Multi-fidelity search
- Warm-start search
- Large number of evaluations
- Nasty objective functions: lots of local optimals, non-stationality
- Indirect observation

Batch / Parallel Bayesian optimization



Batch / Parallel Bayesian optimization

Common approaches:

- Draw multiple independent samples Thomas Sampling.
- Extend acquisition functions with multiple points
- Penalize an acquisition function near selected locations.

Multiple Independent Thomas Samples

[Hernández-Lobato et al., 2017]

Algorithm 2 Parallel and distributed Thompson sampling

Input: initial data $\mathcal{D}_{\mathcal{I}(1)} = \{\mathbf{x}_i, y_i\}_{i \in \mathcal{I}(1)}$, batch size S

for $t = 1$ **to** T **do**

  Compute current posterior $p(\boldsymbol{\theta} | \mathcal{D}_{\mathcal{I}(t)})$

  for $s = 1$ **to** S **do**

    Sample $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta} | \mathcal{D}_{\mathcal{I}(t)})$

    Select $k(s) \leftarrow \text{argmax}_{j \notin \mathcal{I}(t)} \mathbf{E}[y_j | \mathbf{x}_j, \boldsymbol{\theta}]$

    Collect $y_{k(s)}$ by evaluating f at $\mathbf{x}_{k(s)}$

    Executed in parallel in node s

  end for

  $\mathcal{D}_{\mathcal{I}(t+1)} = \mathcal{D}_{\mathcal{I}(t)} \cup \{\mathbf{x}_{k(s)}, y_{k(s)}\}_{s=1}^{S}$

end for

q-EI, q-UCB

[Ginsbourger et al., 2007], [Wang et al., 2016], [Wilson et al., 2018]

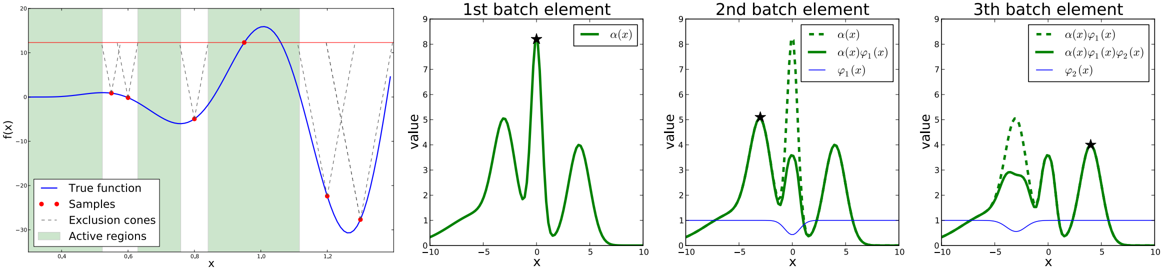
The multi-point extension of EI:

$$q\text{-EI}(\mathbf{X}) = \mathbb{E}\left[\max(f^* - \min_{i=1,\ldots,q} f(\mathbf{x}_i), 0)\right]$$

Local Penalization

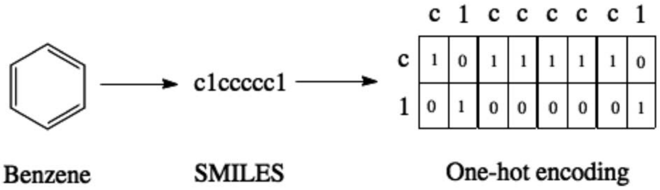
[González et al., 2016]

Assume the object function is Lipschitz continuous.



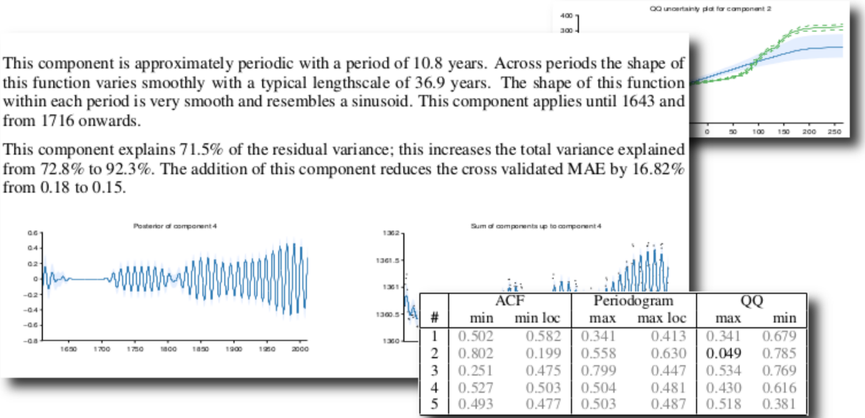
Search in Structured Space: Chemical design

Chemical design for generating novel molecules with optimized properties



Search in Structured Space: Automatic statistician

Automatic exploratory data analysis



This component is approximately periodic with a period of 10.8 years. Across periods the shape of this function varies smoothly with a typical lengthscale of 36.9 years. The shape of this function within each period is very smooth and resembles a sinusoid. This component applies until 1643 and from 1716 onwards.

This component explains 71.5% of the residual variance; this increases the total variance explained from 72.8% to 92.3%. The addition of this component reduces the cross validated MAE by 16.82% from 0.18 to 0.15.

#	ACF		Periodogram		QQ	
	min	min loc	max	max loc	max	min
1	0.502	0.582	0.341	0.413	0.341	0.679
2	0.802	0.199	0.558	0.630	0.049	0.785
3	0.251	0.475	0.799	0.447	0.534	0.769
4	0.527	0.503	0.504	0.481	0.430	0.616
5	0.493	0.477	0.503	0.487	0.518	0.381

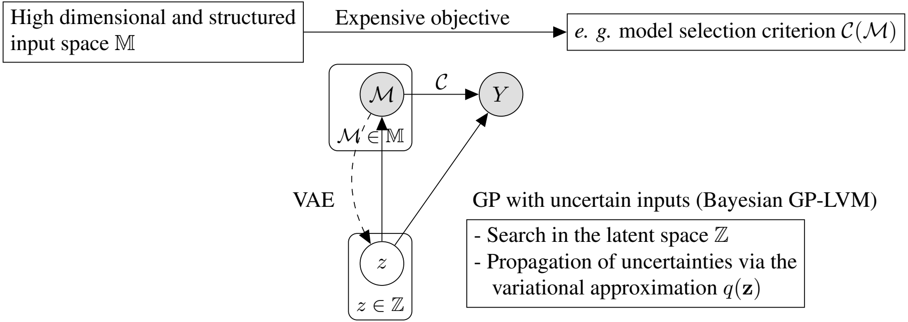
Embed Structured Space into Latent Space

- Map a fixed length continuous representation into a structured (varying length) representation.
- Learning such a mapping from data with probabilistic generative models such as VAE:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}$$

Bayesian Optimization in Latent Space

- [Lu et al., 2018] Encode the known grammar into VAE.
- [Griffiths and Hernández-Lobato, 2020] Learn the grammar from data.

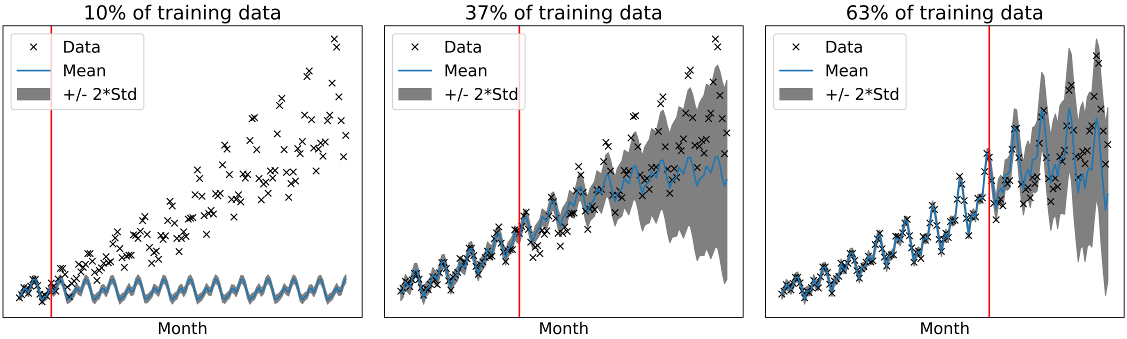


Grammar-based kernel representation

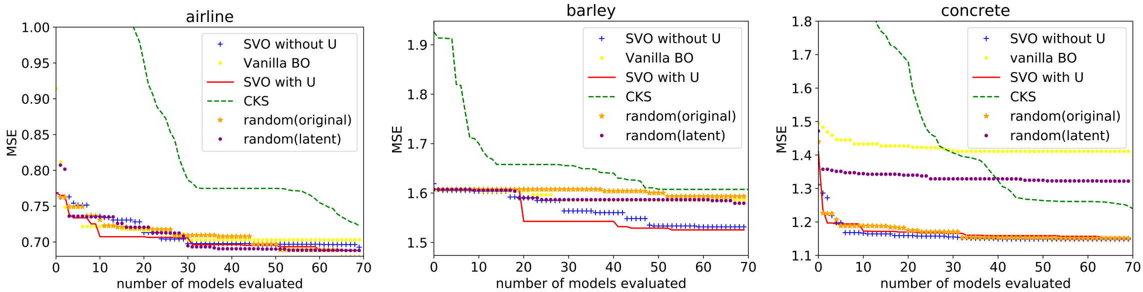
- For regression, it is formulated as model selection on an finite combinatorial space of kernel compositions.
- A set of basic kernels: linear, stationary, period, ...
- The sum or product of two kernels is still a kernel.
- A grammar-based kernel representation

$$\underbrace{K_2}_{0100}\ \underbrace{+}_{100}\ \underbrace{K_1}_{1000}\ \underbrace{*}_{010}\ \underbrace{K_3}_{0010}\ \underbrace{*}_{010}\ \underbrace{K_1}_{1000}\ \underbrace{Stop}_{001}\ \underbrace{\ldots}_{Add\ 000}$$

Experiment results on automatic statistician



Experiment results on automatic statistician

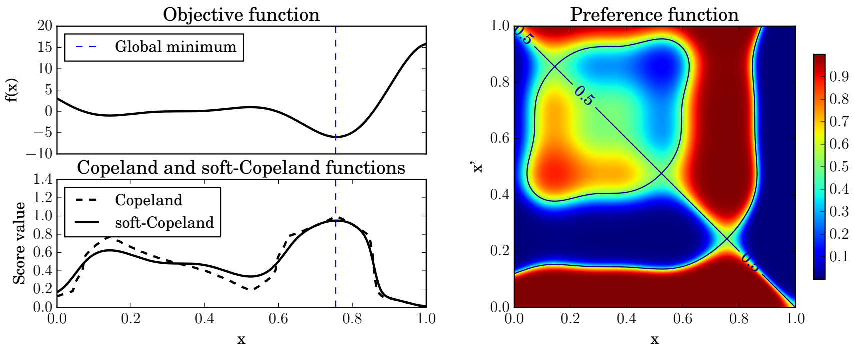


Preferential Bayesian Optimization

- Many functions that we are interested in optimizing is hard to measure:
  - user experience, e.g, UI design
  - movie/music rating
- Human are much better at comparing two things, e.g., is this coffee better than the previous one?
- To search for the most preferred option via only pair-wise comparisons.

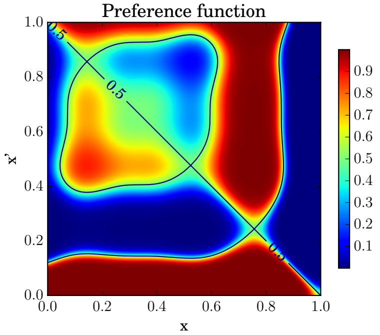
Preference Function

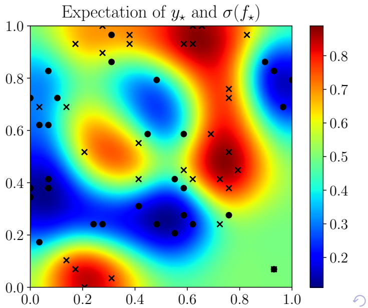
- Preference function: $p(y = 1|x, x') = \pi(x, x') = \sigma(g(x') - g(x))$.
- Copeland function: $S(x) = \frac{1}{\mathrm{Vol}(\mathcal{X})} \int_{\mathcal{X}} \mathbb{I}_{\pi(x,x') \geq 0.5} \mathrm{d}x'$.
- The minimal of a Copeland function corresponds to the most preferred choice.



A Surrogate Model of Preference Function

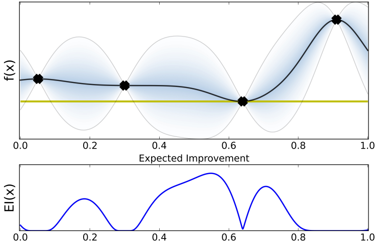
- The preference function is not observable.
- Only observe a few comparisons.
- Need a surrogate model to guide the search.
- We propose to build a surrogate model for the preference function.
- Pros: easy to model (Gaussian process Binary Classification is used:)
- Pros: flexible latent function (e.g., non-stationality).
- Cons: the minimum of the latent function is not directly accessible

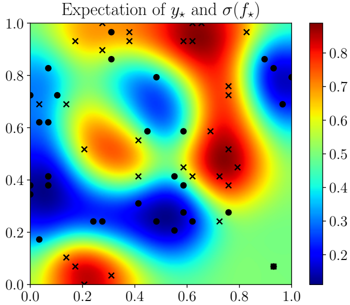




Acquisition Function

- Existing Acq. Func. are not *applicable*.
- They are designed to work with a surrogate model of the objective function.
- In PBO, the surrogate model does not directly represent the *latent* objective function.
- We need a new Acq. Func. for duels!

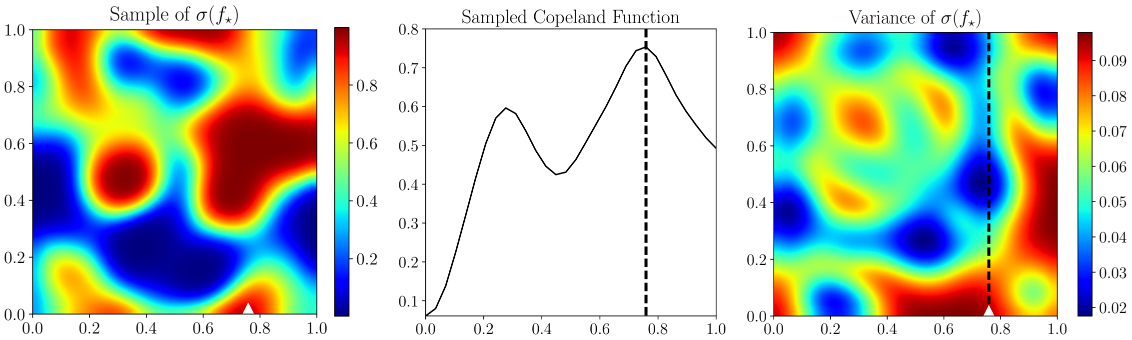




Acquisition Function: PBO-DTS

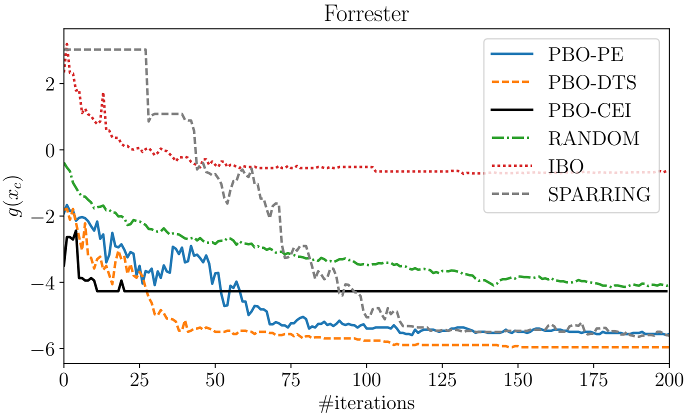
To select the next duel $[x, x']$:

1. Draw a sample from surrogate model
2. Take the maximum of *soft-Copeland* score as x.
3. Take x' that gives the maximum in PBO-PE



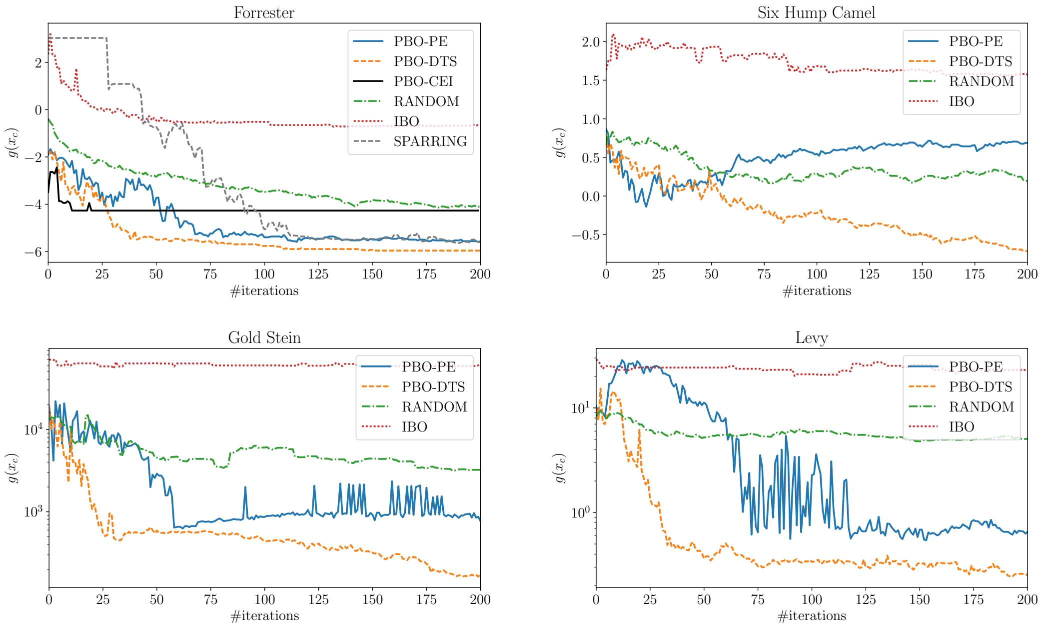
Experiment: Forrester Function

- Synthetic 1D function: Forrester
- Observations drawn with a probability: $\frac{1}{1+e^{g(x)-g(x')}}$
- $g(x_c)$ shows the value at the location that algorithms *believe* is the minimum.
- The curve is the average of 20 trials.



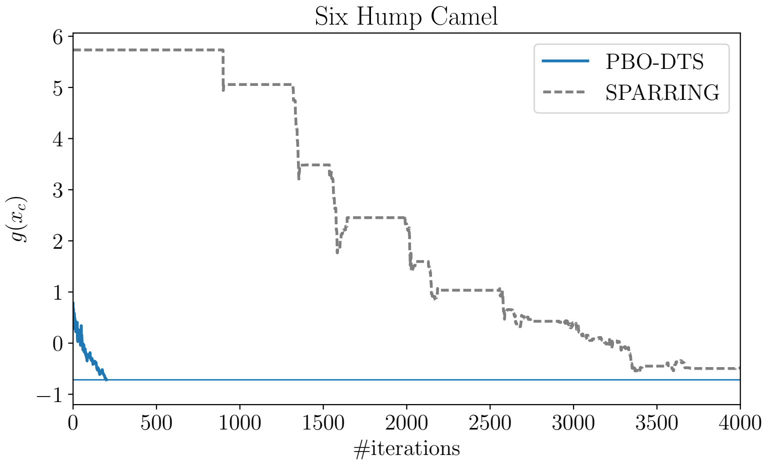
IBO: [Brochu, 2010]
SPARRING: [Ailon et al., 2014]

Experiments: More (2D) Functions

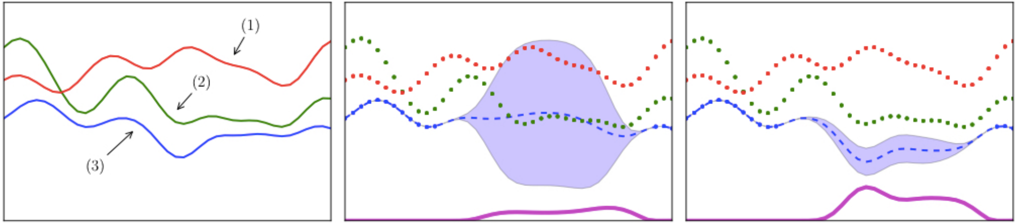


When no correlation considerred

Discretize the 2D space into a 30 x 30 grid and apply dueling bandits.



Multi-task Bayesian optimization



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

Relation to Hyper-Parameter Optimization (HPO)

- HPO is a concrete global optimization problem with expensive objective functions.
- BO has been a very successful method for HPO.
- There are many other optimization methods developed dedicated to HPO such as Hyperband, BOHB.

Q & A

Nir Ailon, Zohar Shay Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 856–864, 2014.

Eric Brochu. *Interactive Bayesian Optimization: Learning Parameters for Graphics and Animation*. PhD thesis, University of British Columbia, Vancouver, Canada, December 2010.

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. A multi-points criterion for deterministic parallel global optimization based on kriging. 2007.

J. González, Z. Dai, P. Hennig, and N. Lawrence. Batch bayesian optimization via local penalization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 648–657, 2016.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.*, 11:577–586, 2020.

José Hernández-Lobato, James Requeima, Edward Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. In *ICML*, 2017.

Xiaoyu Lu, Javier Gonzalez, Zhenwen Dai, and Neil Lawrence. Structured variationally auto-encoded optimization. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

J. Močkus. On bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404. Springer Berlin Heidelberg, 1975.

Jialei Wang, Scott Clark, Eric Liu, and Peter Frazier. Parallel bayesian global optimization of expensive functions. 2016.

JT Wilson, F Hutter, and Marc Deisenroth. Maximizing acquisition functions for bayesian optimization. In *Neurips*, 2018.