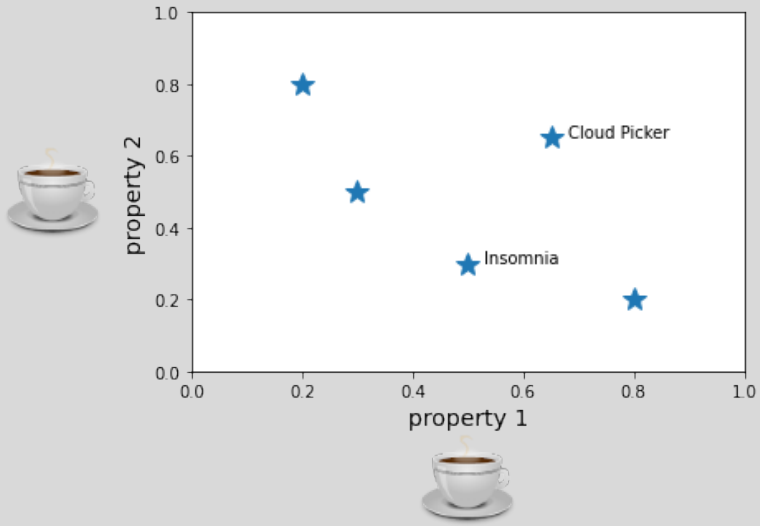
Bayesian Optimisation with preference and choice data with application to smart manufacturing

Alessio Benavoli
Associate Professor
Discipline of Statistics and Information Systems
SCSS, Trinity College Dublin
alessio.benavoli@tcd.ie



Overview

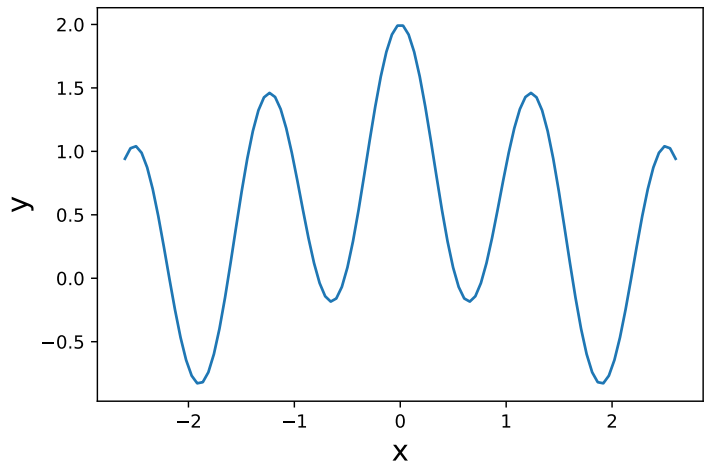
1. Bayesian Optimisation
2. Bayesian Preferential Optimisation
3. Batch Bayesian Preferential Optimisation
4. Bayesian Choice Optimisation

Bayesian Optimisation

What is BO?

It is a methodology for global black-box optimisation of functions that are expensive to evaluate.

Imagine we want to find the maximum of this 1D function:

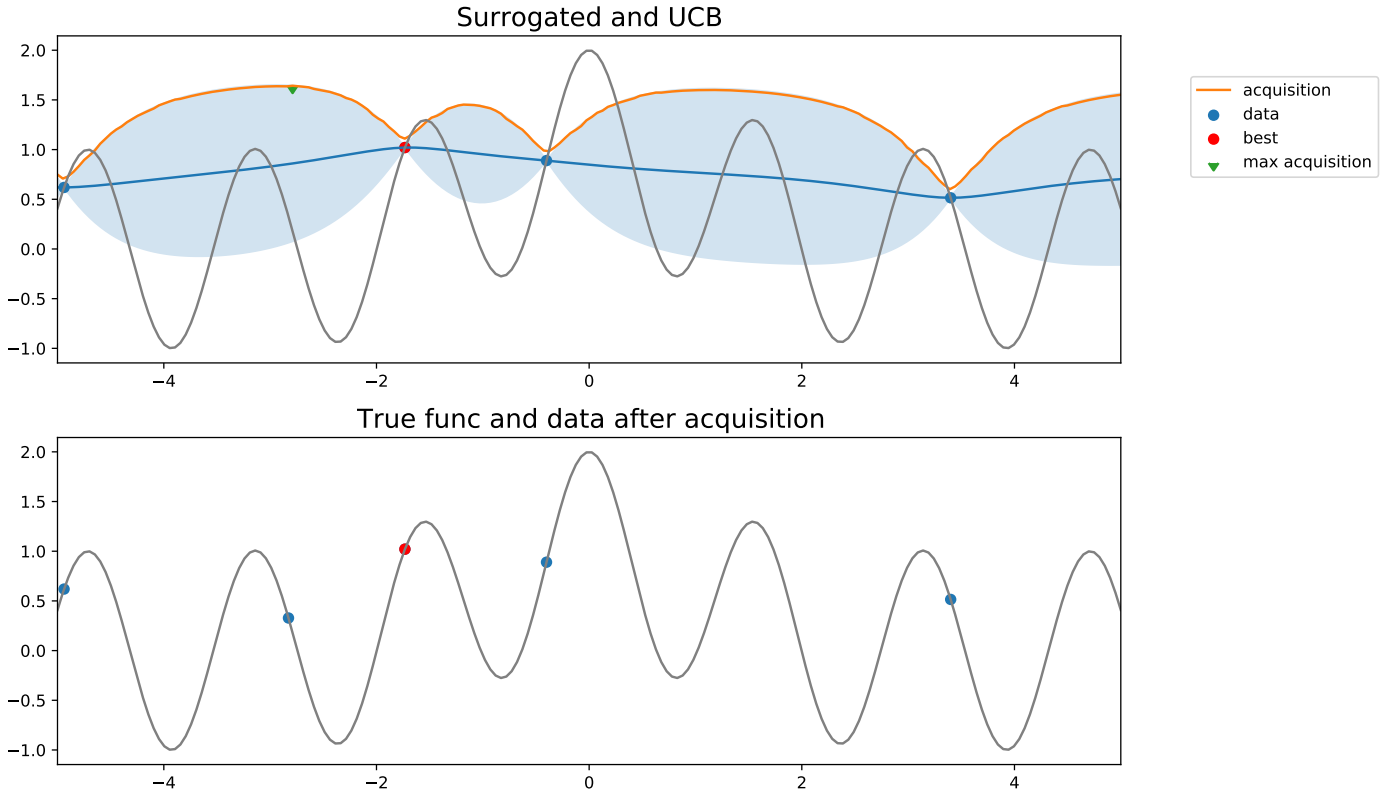


but we do not know the function, that is we can only evaluate it.

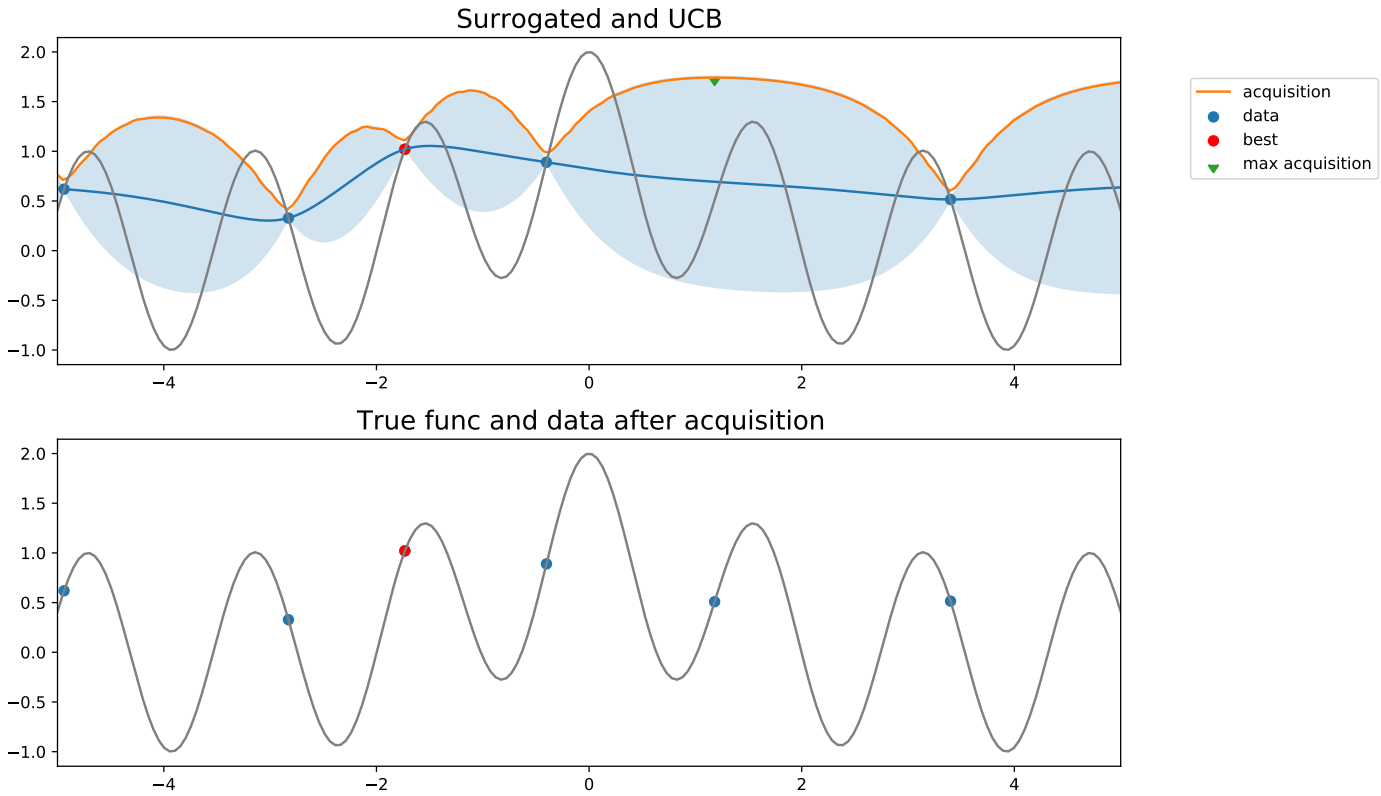
BO loop

We start from some initial data points data $= \{(x_i, g(x_i)) \text{ for } i = 1, 2, 3\}$.
Loop:

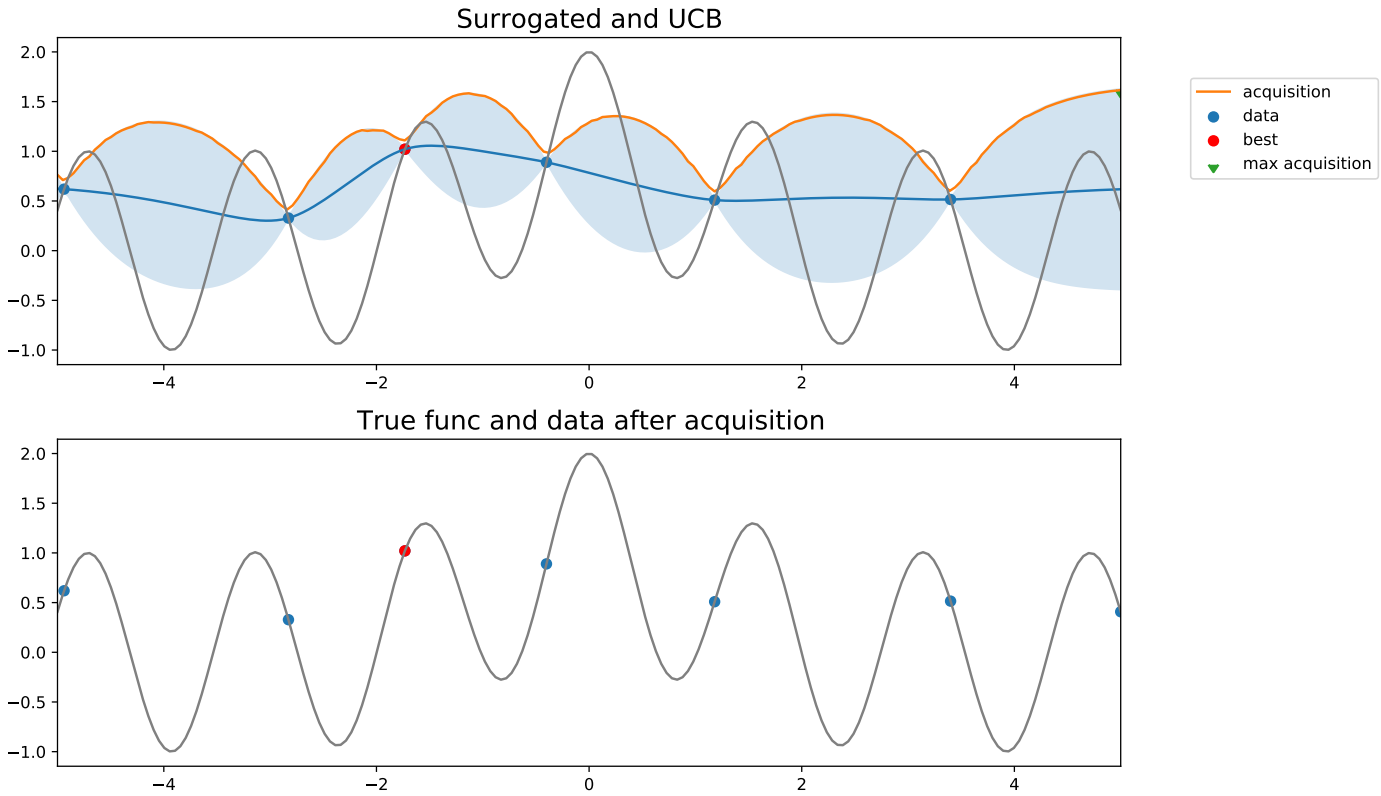
1. compute a GP regression model (surrogate model);
2. optimise an acquisition function to compute x_{next}
3. evaluate g at x_{next} and update data $=$ data $\cup \{(x_{next}, g(x_{next}))\}$



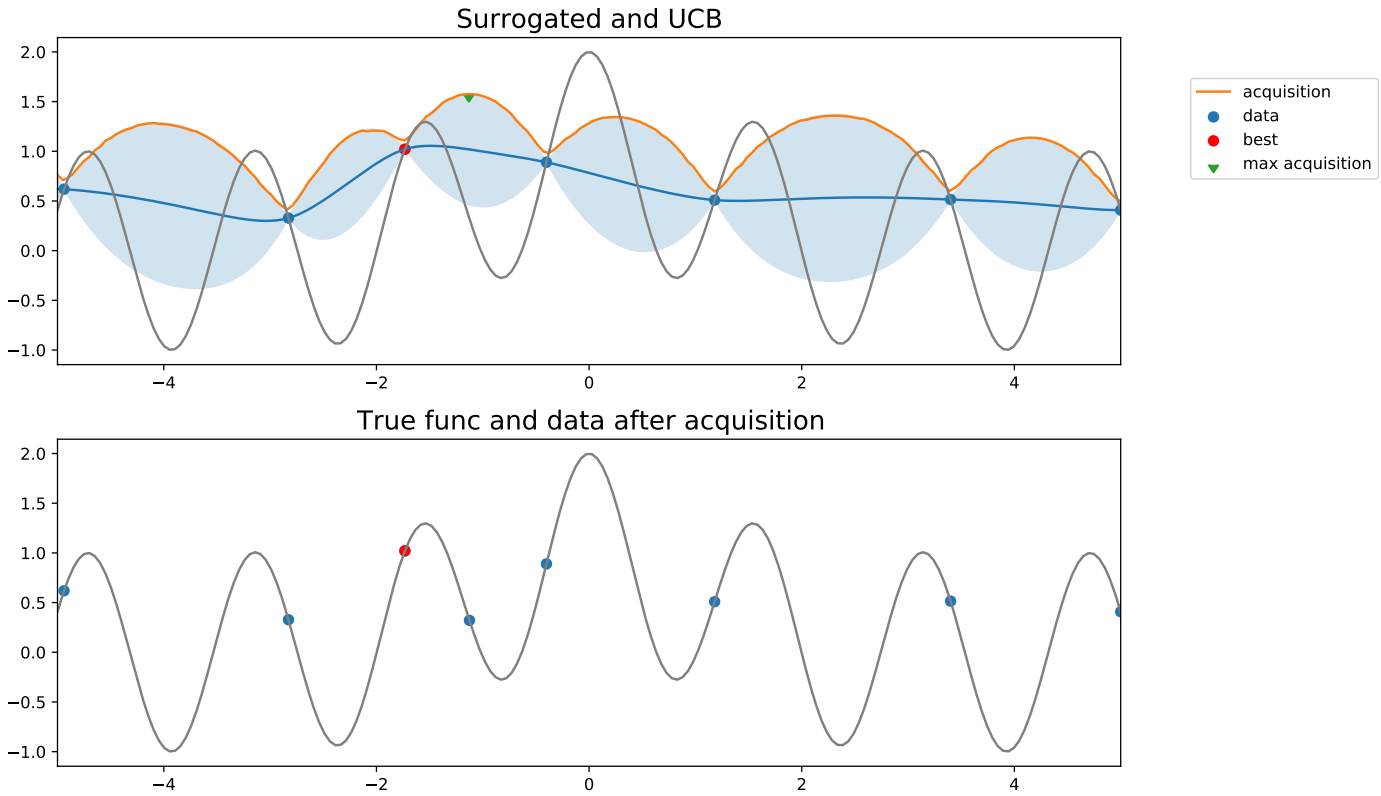
BO loop


Surrogated and UCB
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4
acquisition
data
best
max acquisition
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

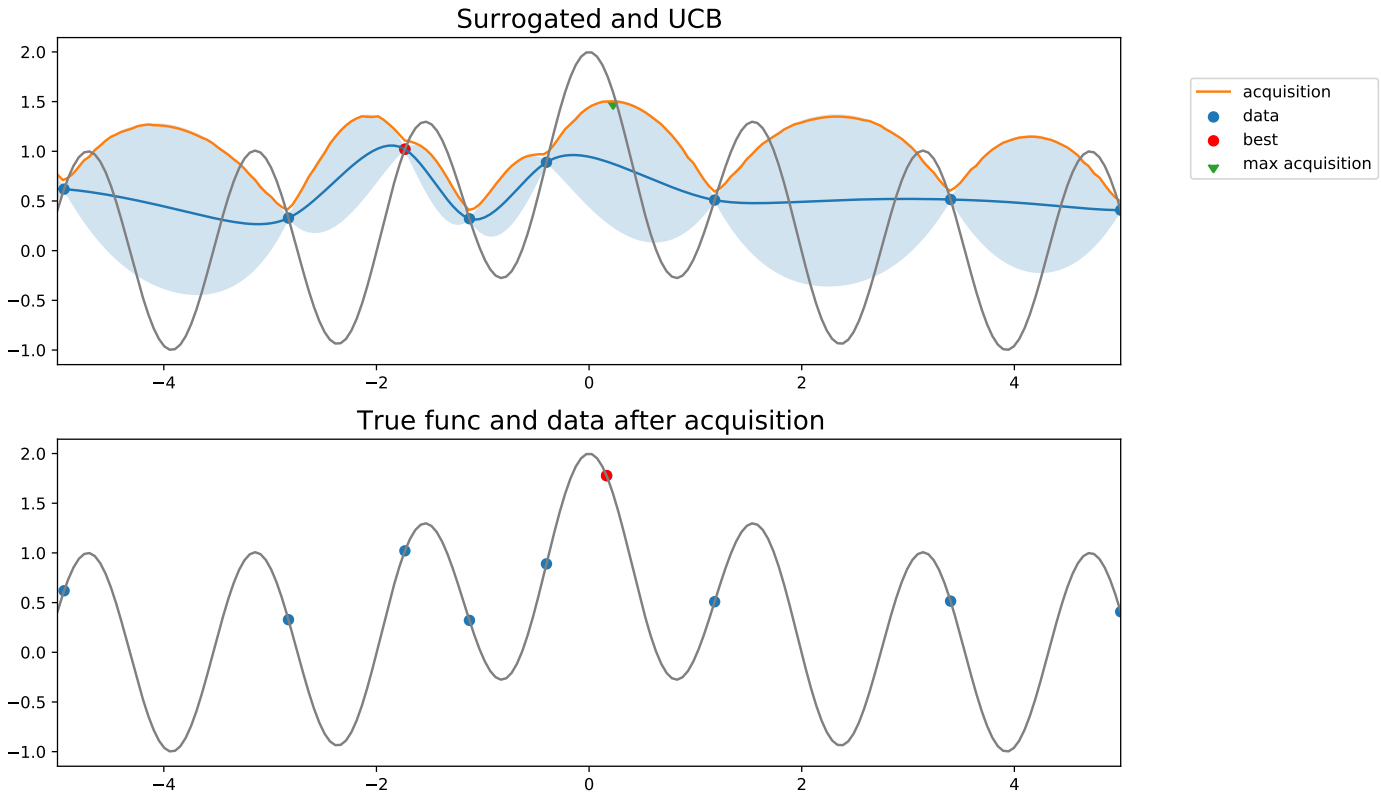
BO loop


Surrogated and UCB
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4
acquisition
data
best
max acquisition
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

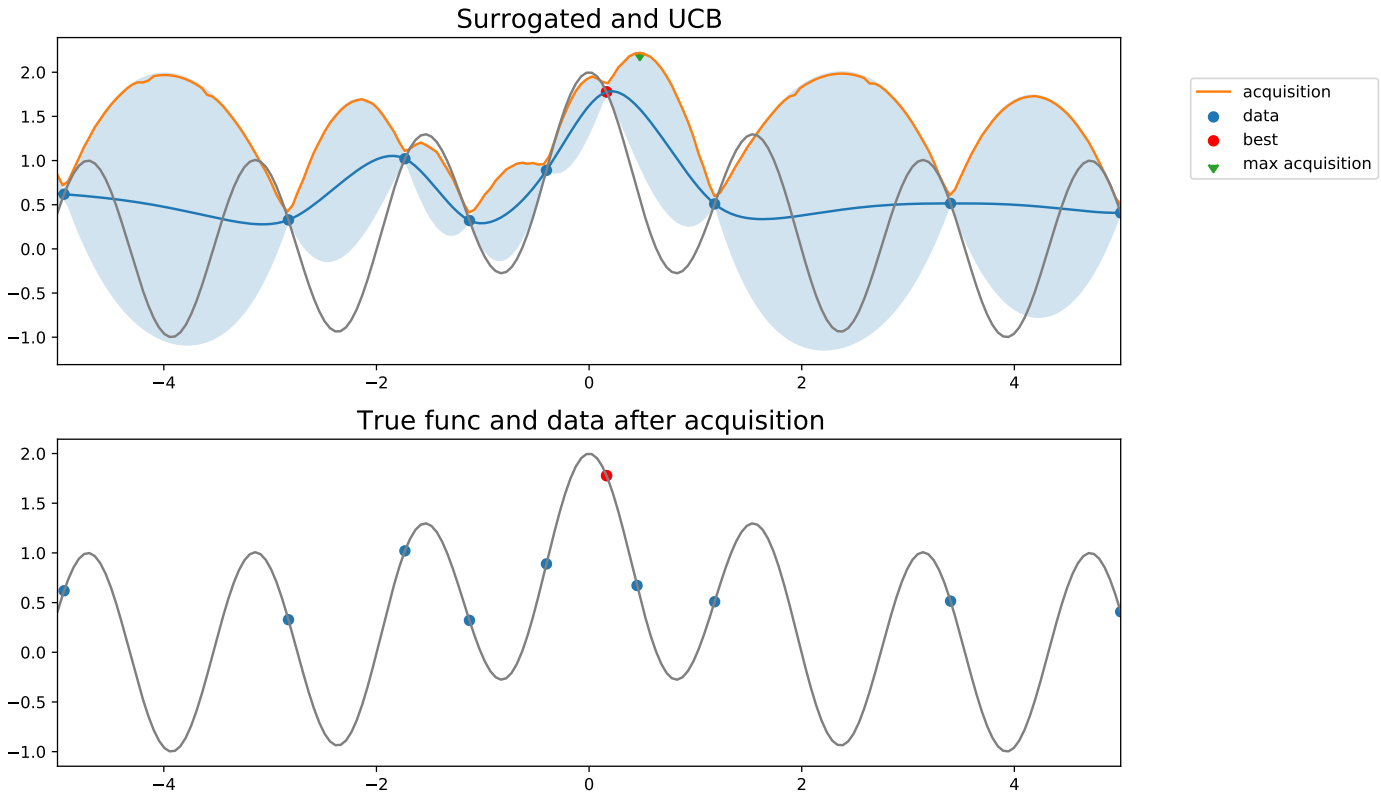
BO loop


Surrogated and UCB
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4
acquisition
data
best
max acquisition
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

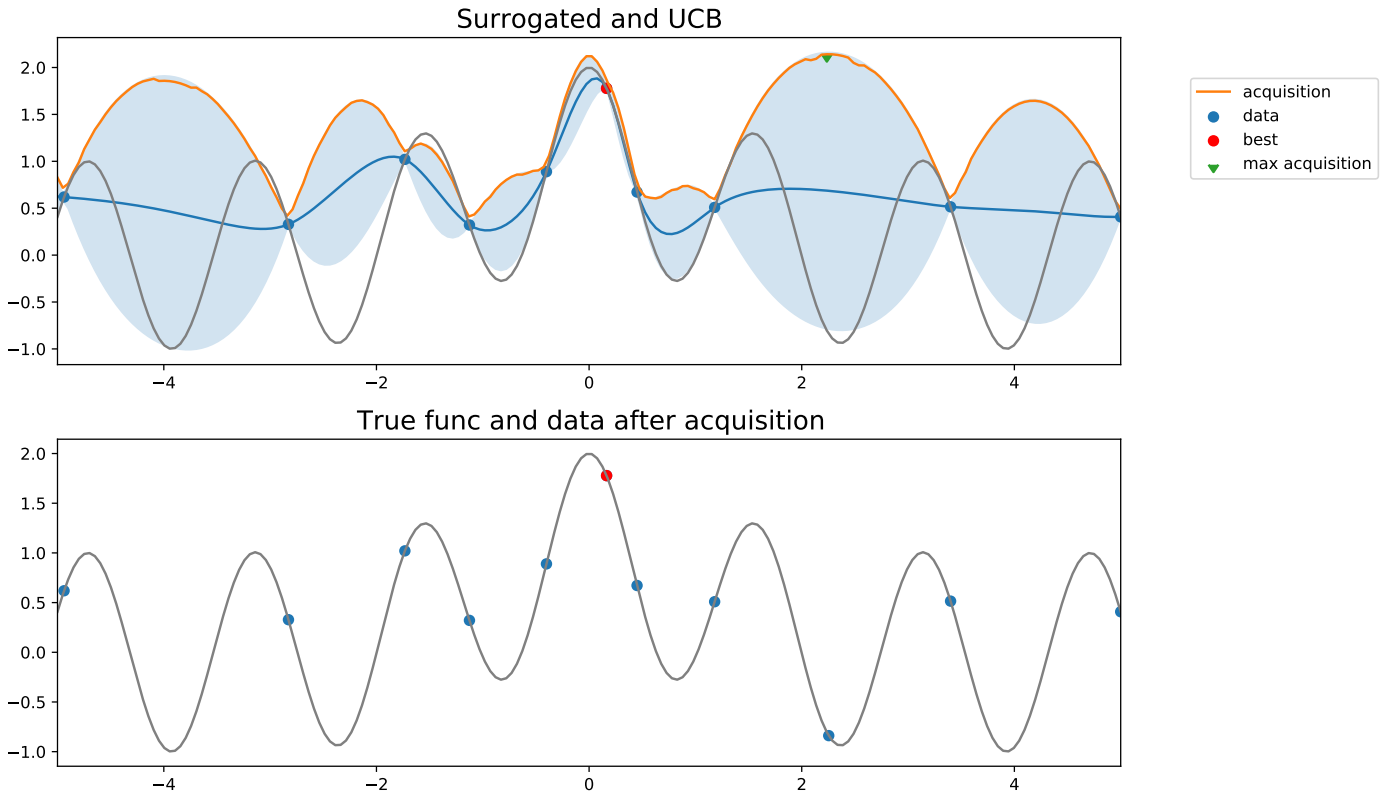
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

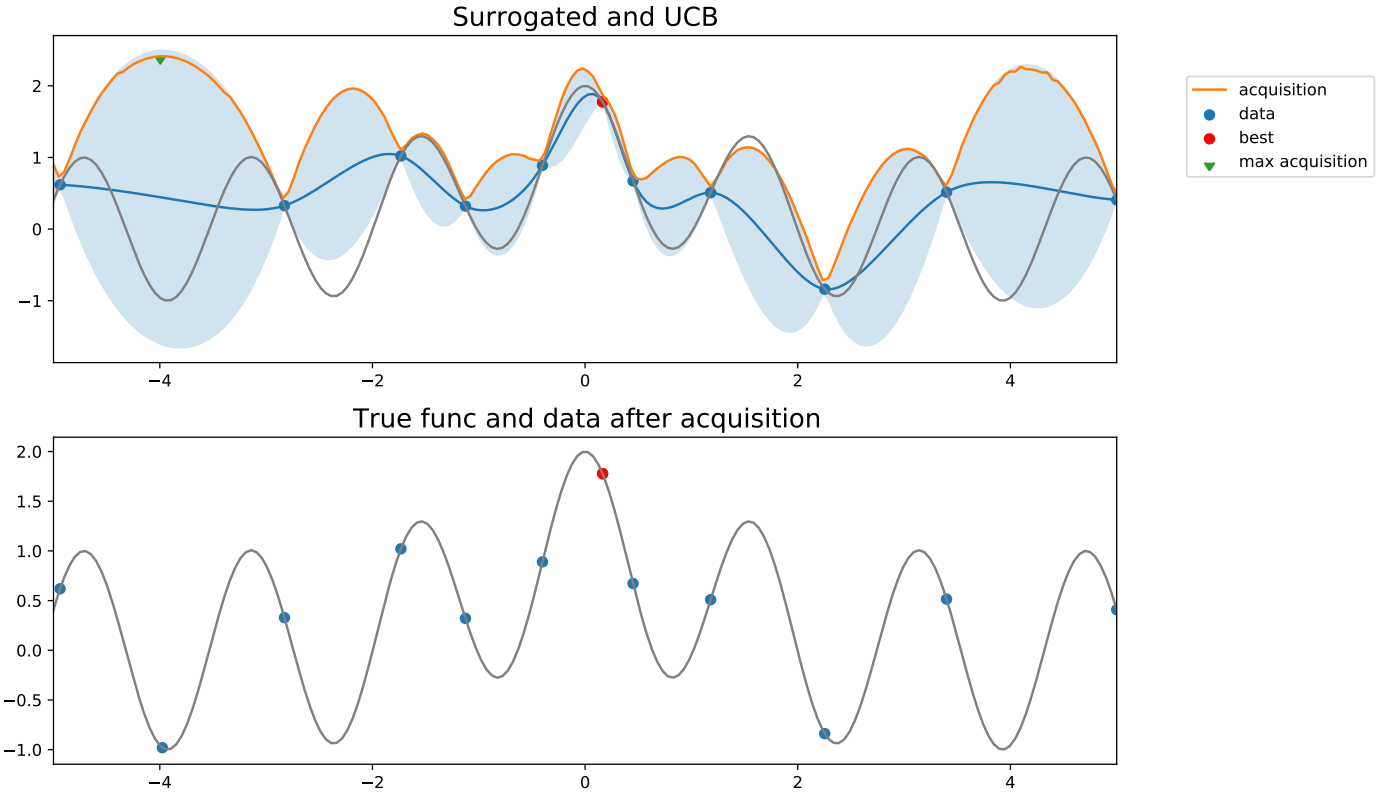
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

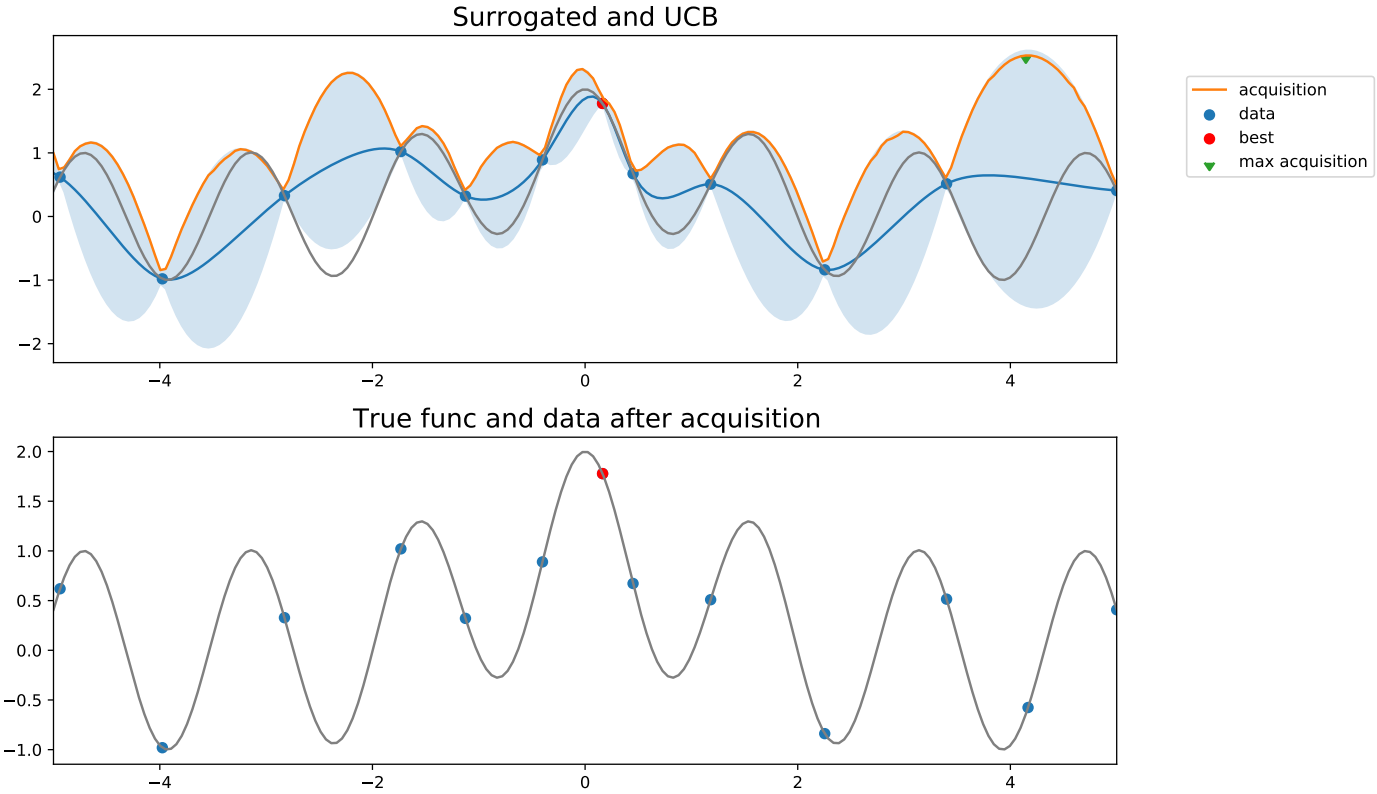
BO loop


Surrogated and UCB
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4
acquisition
data
best
max acquisition
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

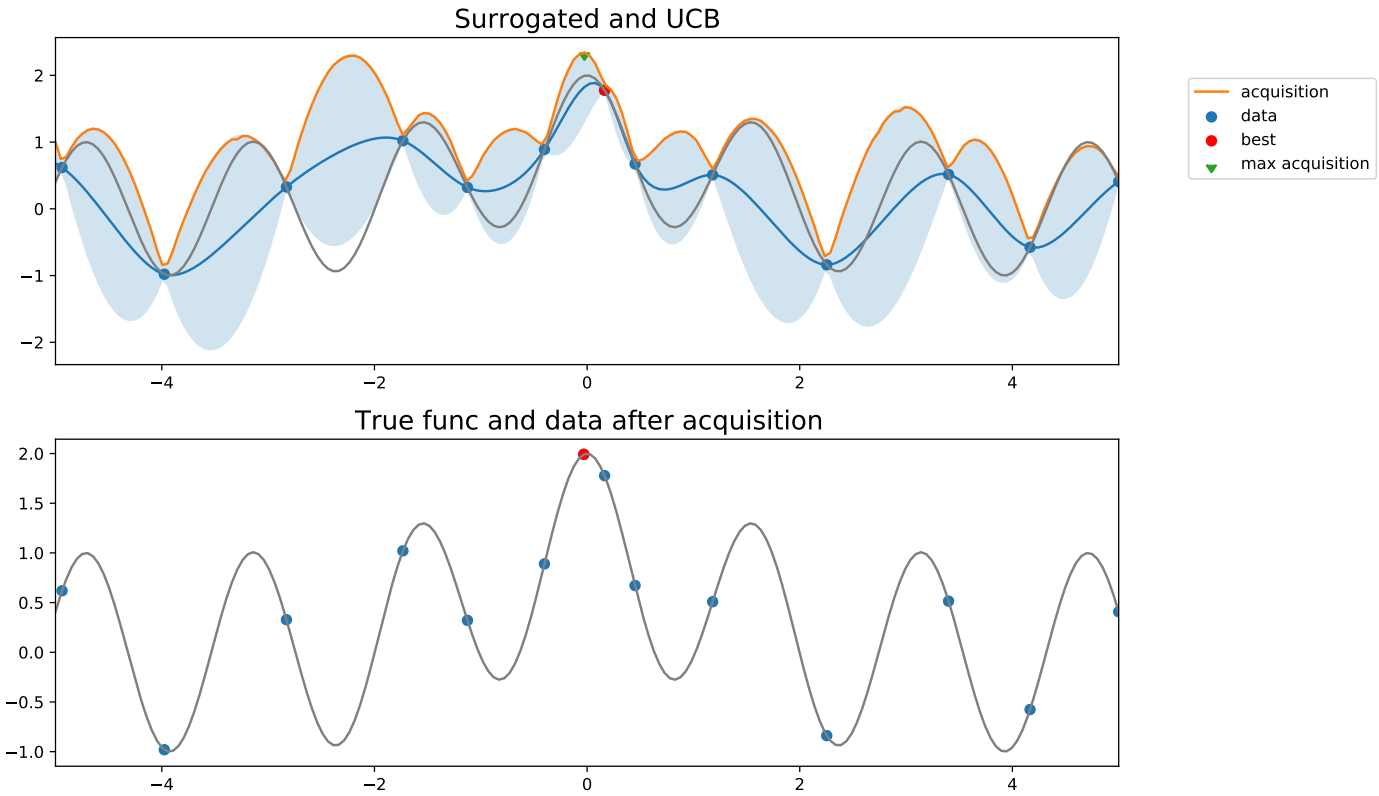
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

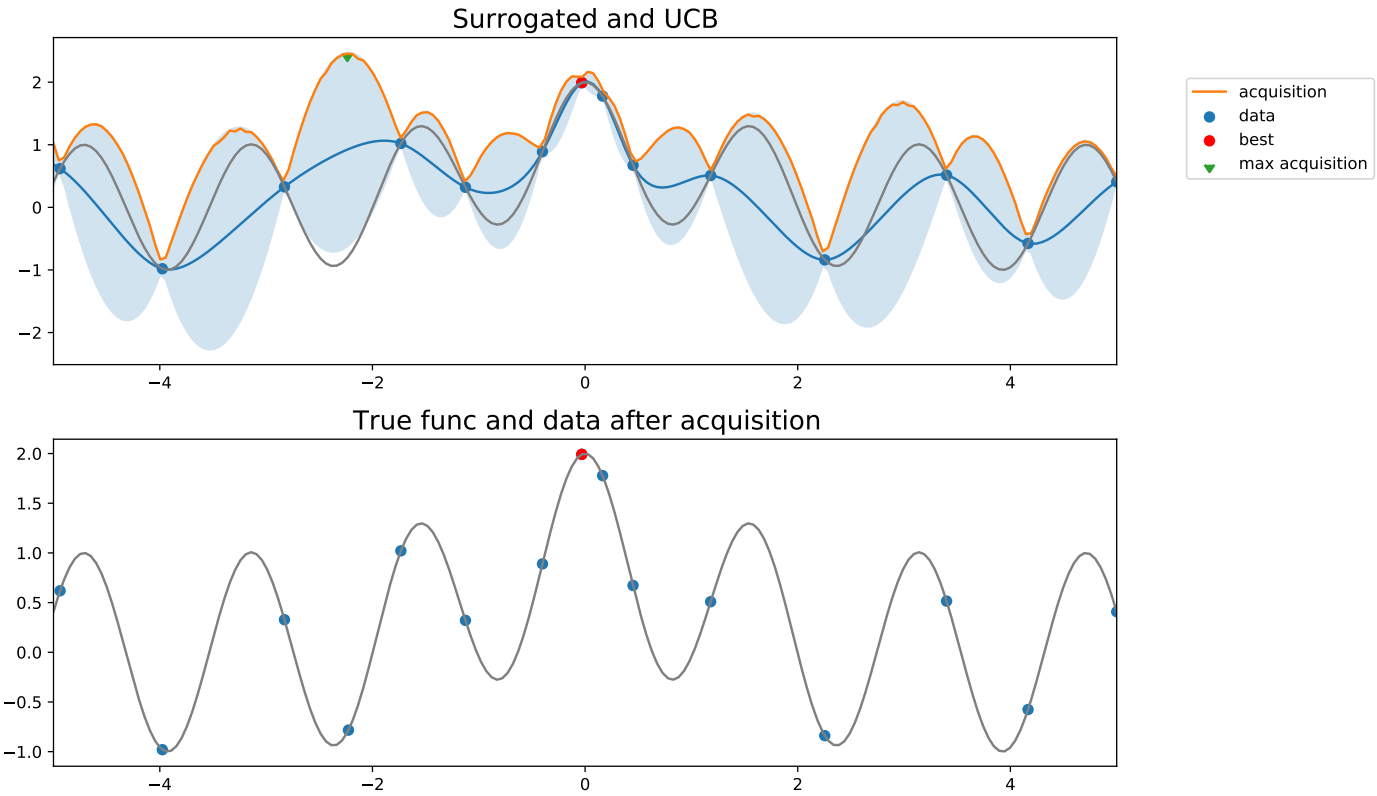
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

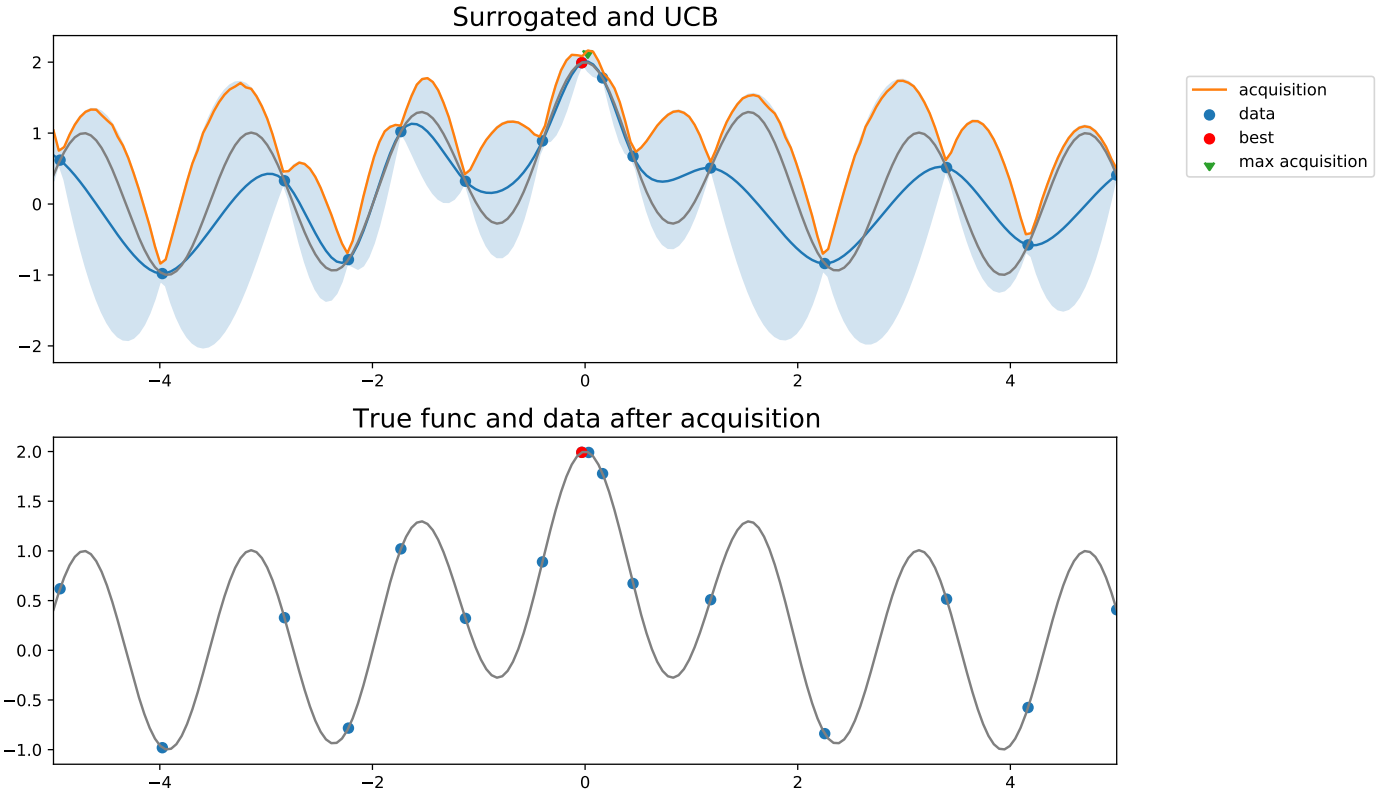
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

BO loop

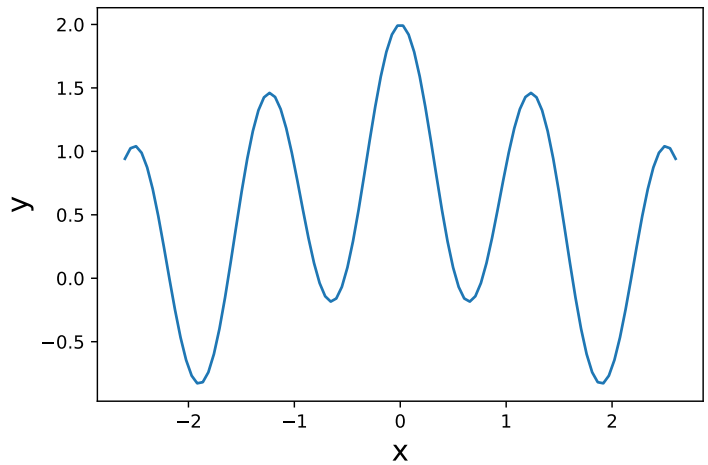

Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

Bayesian Optimisation (noise)

What is BO?

It is a methodology for global black-box optimisation of functions that are expensive to evaluate.

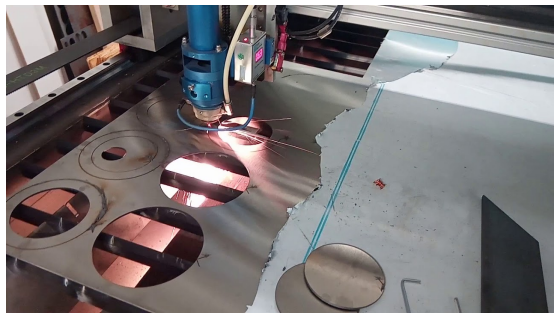
Imagine we want to find the maximum of this 1D function:



but we do not know the function: we can only collect noisy evaluations of it.

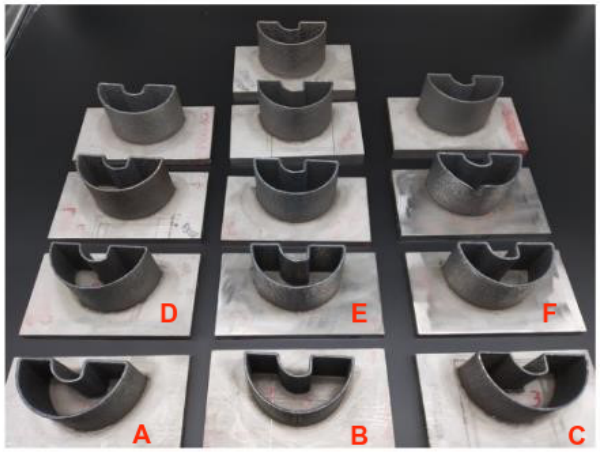
Application: Smart Manufacturing

- Metal Additive Manufacturing;
- Metal Laser Cutting;
- Metal Electrical Discharge Machine;



Goal: find the parameters (setting) of the machine that optimise quality/speed/reliability

Additive Manufacturing



The process can be represented as

$$\text{roughness} = g(\text{laser power}, \text{scan speed}, \ldots) + \text{noise}$$

and the goal is:

$$\min_{\text{laser power},\text{scan speed},\ldots} E[g(\text{laser power}, \text{scan speed}, \ldots)]$$

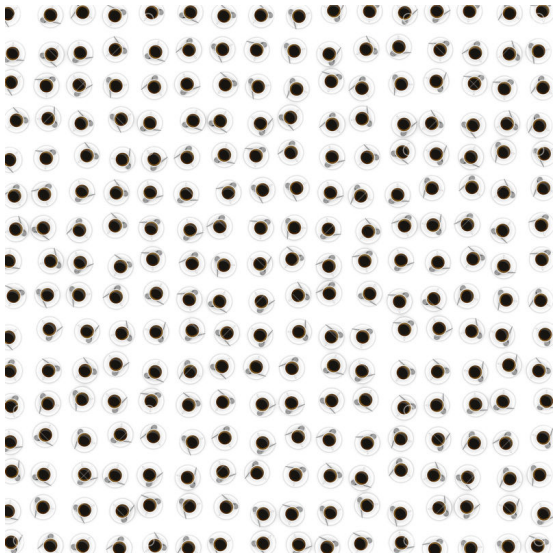
Issue: measuring quantitatively roughness is difficult and costly (time, microscope)

Fact: an expert user is able to optimise the parameters of the machine by simply looking at/touching the parts

BO for coffee machine

Manufacturing process:

DOE (full factorial)



Setting:

$\mathbf{x} = [\text{temp}, \text{press}, \text{coffee amount}, \text{coffee type}]$

we aim to make the best coffee.

How do we measure the quality of a cup of coffee?

Bayesian preferential BO

Before we considered the case that

$$\text{ev}_{x_{next}}(g) \rightarrow g(x_{next})$$

but there are situations where evaluating the function is difficult (costly):

$$\text{pref}_{x_{next},x_{best}}(f) = \begin{cases} x_{next} \succ x_{best} & \text{if } g(x_{next}) + \text{noise}_1 > g(x_{best}) + \text{noise}_2 \\ x_{next} \not\succ x_{best} & \text{otherwise} \end{cases}$$

Probabilistic (Surrogate) model:
Assuming independent Gaussian noise $N(0, \sigma^2)$ and a GP prior over the latent function f that determines the preferences:

$$p(f) = GP(0, K)$$

$$p(data|f) = \Phi\left(\frac{f(x_{next}) - f(x_{best})}{\sigma}\right)^{y} \Phi\left(\frac{f(x_{best}) - f(x_{next})}{\sigma}\right)^{1-y}$$

Posterior

The posterior is not a GP, possible ways to deal with that:

- Laplace approximation
- Expectation Propagation approximation
- Variational approximation
- (Elliptical Slice Sampling (MCMC))

- Chu, W. and Ghahramani, Z. (2005). Preference learning with gaussian processes. In Proceedings of the 22nd International Conference on Machine Learning, ICML '05
- González, J., Dai, Z., Damianou, A., and Lawrence, N. D. (2017). Preferential bayesian optimization. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1282–1291.

- The posterior is a Skew Gaussian Process and we can use analytical derivations to sample from it without rejection via linear elliptical slice sampling

- Benavoli A, Azzimonti D, Piga D. A unified framework for closed-form nonparametric regression, classification, preference and mixed problems with Skew Gaussian Processes. Machine Learning. 2021 Sep 13:1-39.

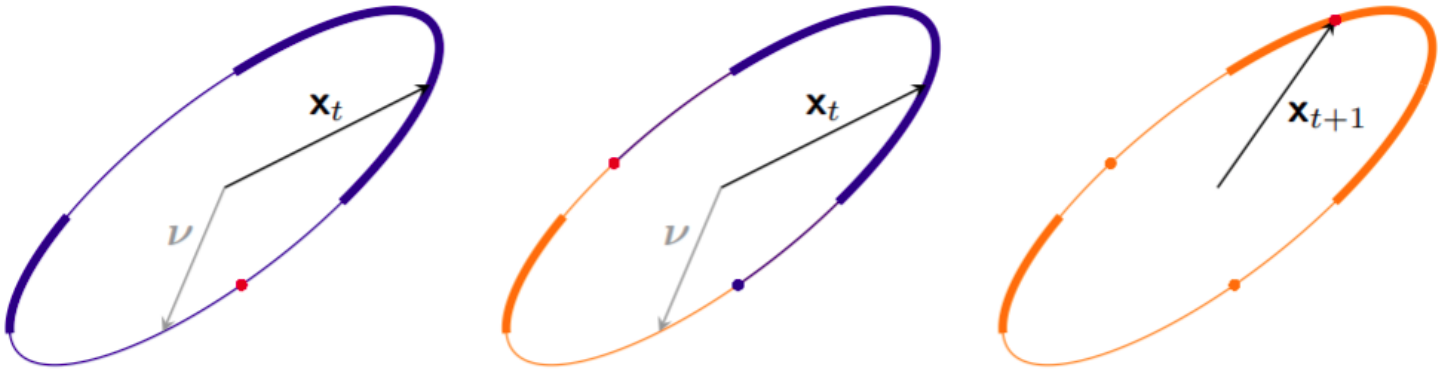
Elliptical Slice Sampling (Murray et al. 2010)

MCMC algorithm for the special case that $p_*(\mathbf{x}) = \ell(\mathbf{x})\,\mathcal{N}(\mathbf{x}, \mathbf{0}, \mathbf{\Sigma})$

Construct **1D** ellipse from state $\mathbf{x}_t$ and auxiliary vector $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{x}, \mathbf{0}, \mathbf{\Sigma})$ as

$$\mathbf{x}(\theta) = \mathbf{x}_t \cos\theta + \boldsymbol{\nu} \sin\theta$$

and perform **slice sampling** on ellipse.



Note: this algorithm is parameter-free!

Linear Elliptical Slice Sampling (Gessner et al. 2019)

Elliptical slice sampling where...

1. "Likelihood" $\ell(\mathbf{x}) = \mathbb{1}_{\mathcal{L}}$ has binary outcome, 0 or 1

   $\rightarrow$ **no likelihood threshold**

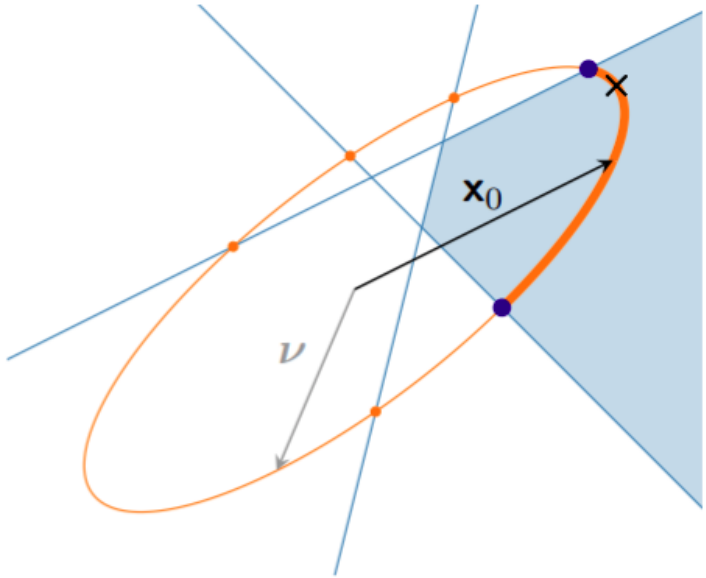
2. Intersections of ellipse and domain boundaries have closed-form solutions

$$\mathbf{A}^\intercal(\mathbf{x}_0 \cos\theta + \boldsymbol{\nu} \sin\theta) + \mathbf{b} = \mathbf{0}$$

$$\theta_{m,1/2} = \pm \arccos\left(-\frac{b_m}{r}\right) + \arctan\left(\frac{\mathbf{a}_m^\intercal \boldsymbol{\nu}}{r + \mathbf{a}_m^\intercal \mathbf{x}_0}\right)$$

with $r = \sqrt{(\mathbf{a}_m^\intercal \mathbf{x}_0)^2 + (\mathbf{a}_m^\intercal \boldsymbol{\nu})^2}$

$\rightarrow$ **rejection-free sampling**

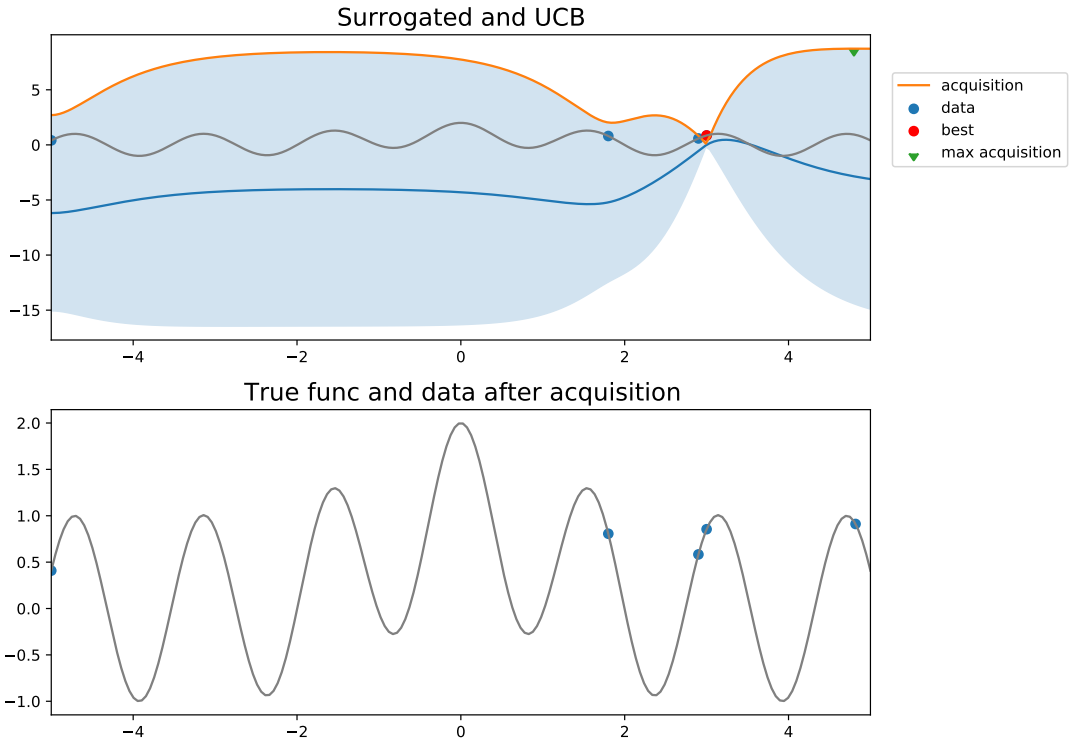


LIN-ESS

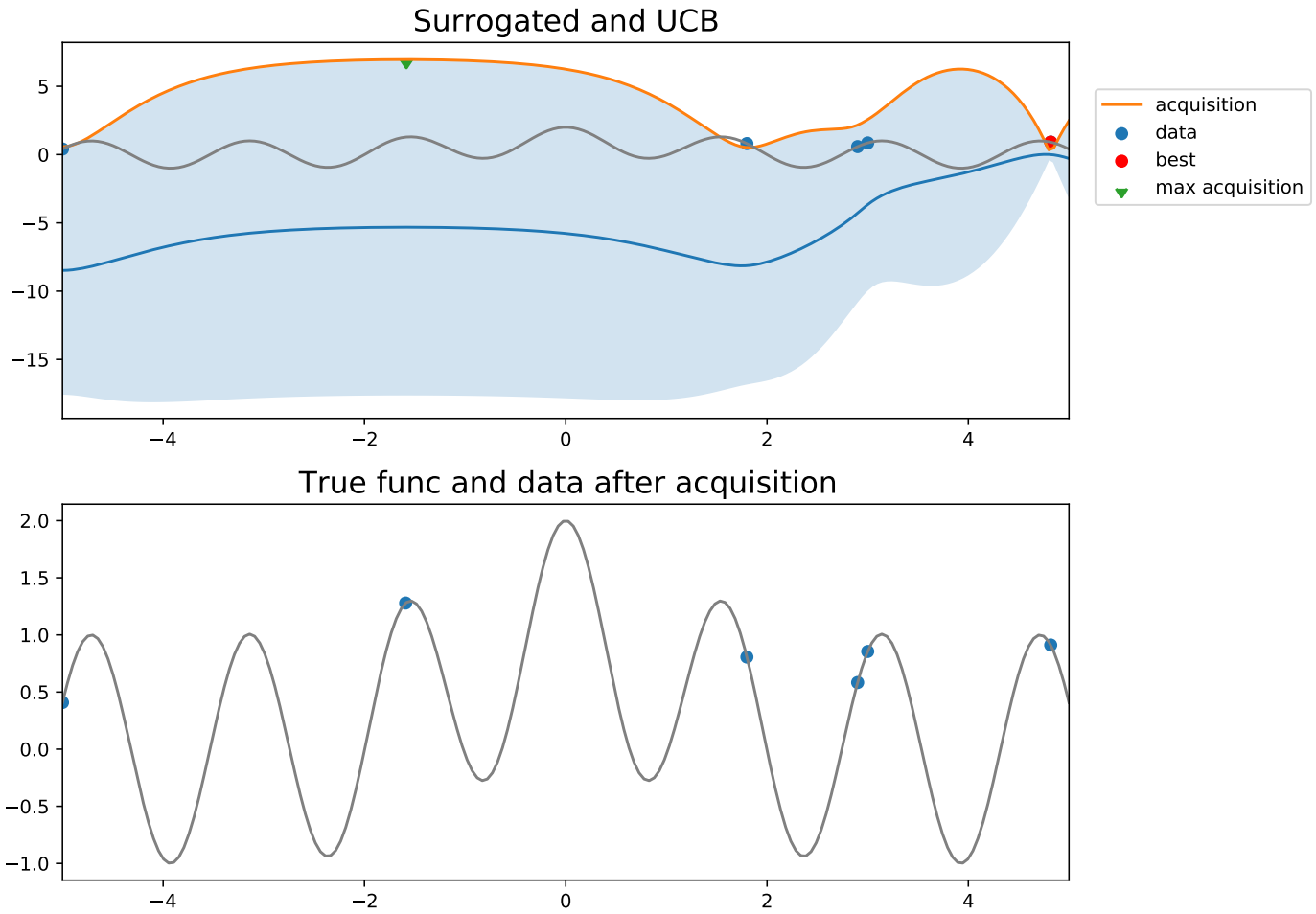
BO preferential loop

We start from some initial data points data $= \{x_1 \succ x_2, x_1 \succ x_3\}$.
Loop:

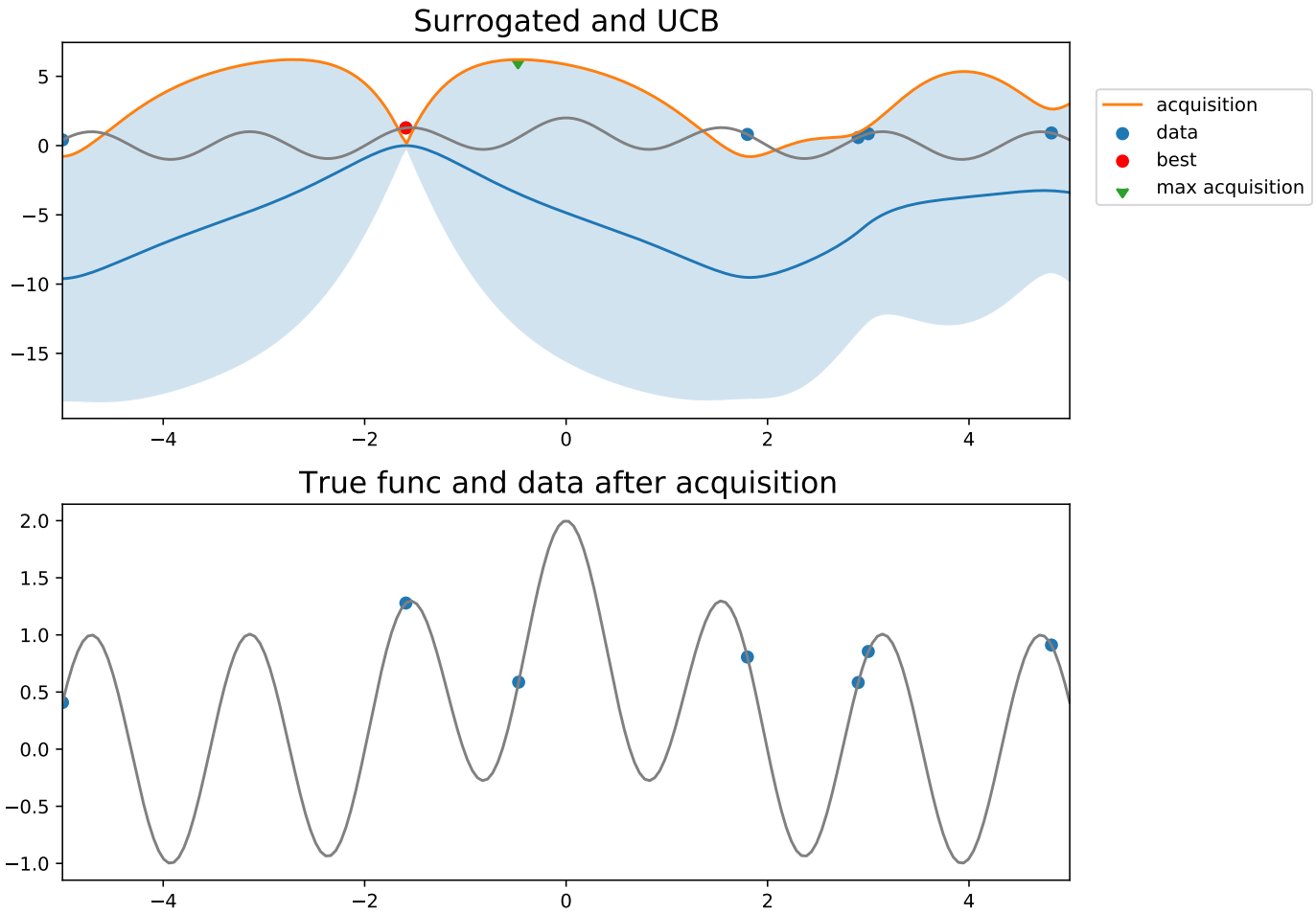
1. learn a surrogate model (SkewGP);
2. optimise an acquisition function to compute x_{next}
3. query f for x_{next} versus x_{best} and update data $=$ data $\cup \{x_{next} \succ x_{best}\}$



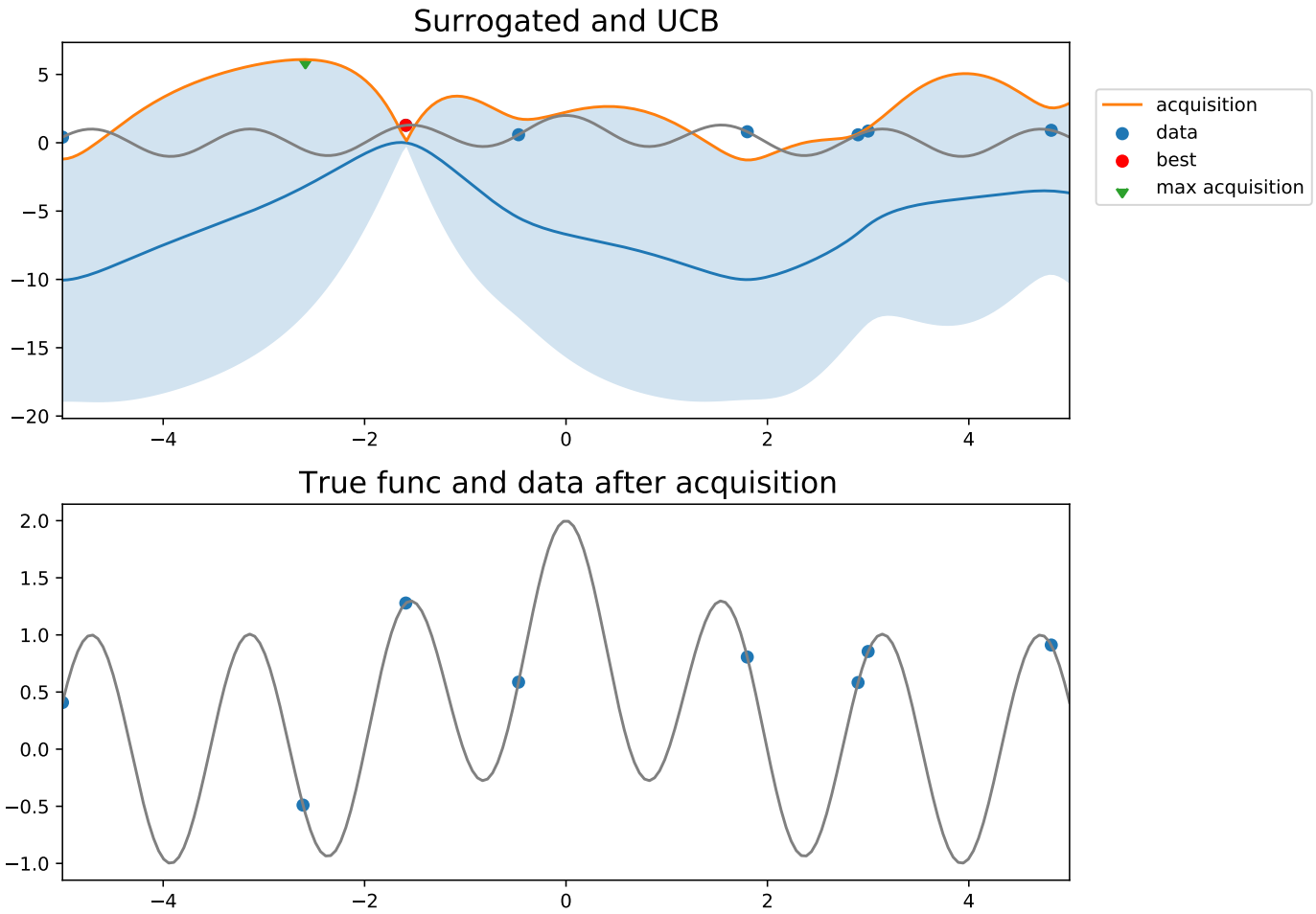
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

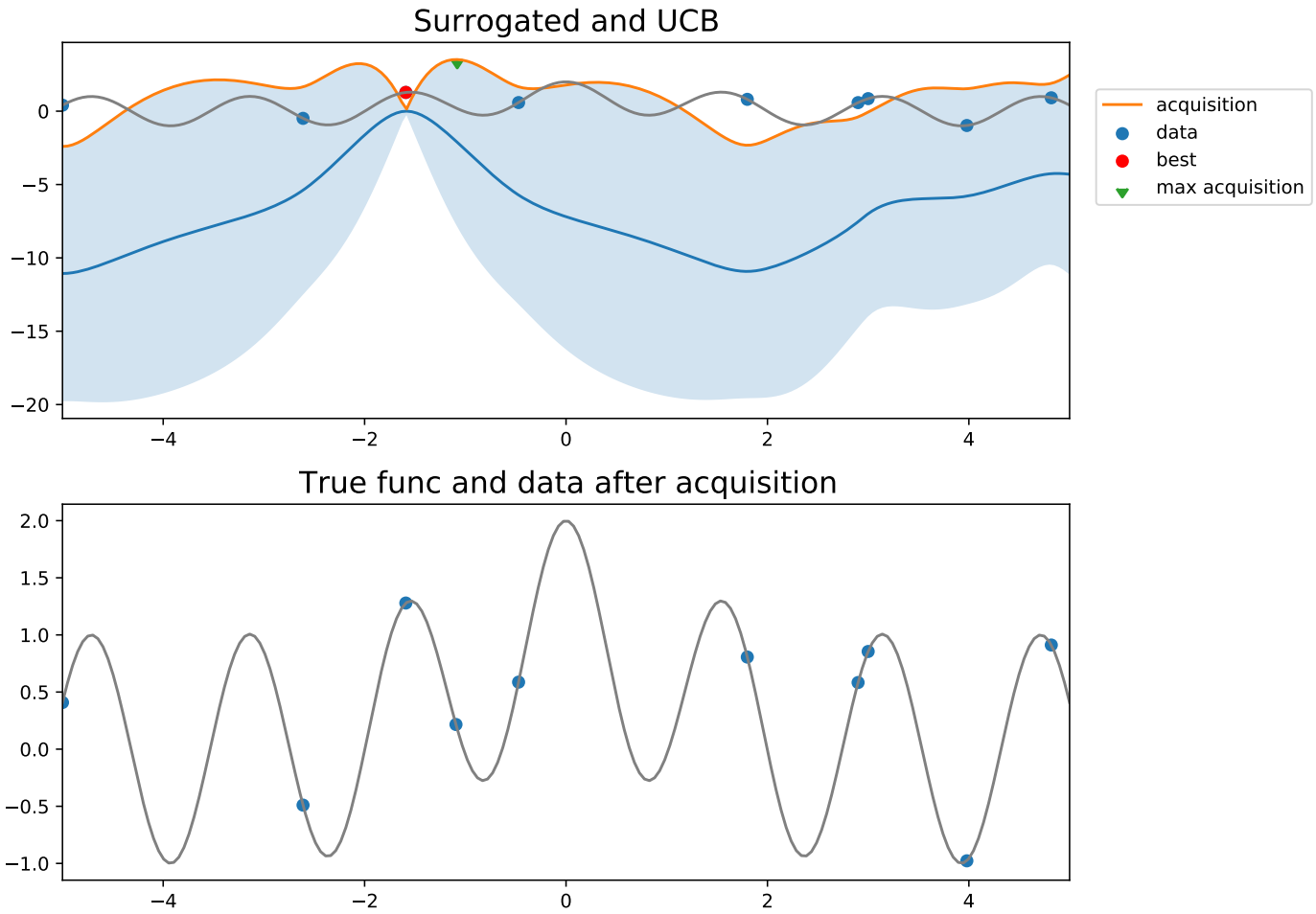
BO loop


Surrogated and UCB
5
0
−5
−10
−15
−4
−2
0
2
4
acquisition
data
best
max acquisition
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

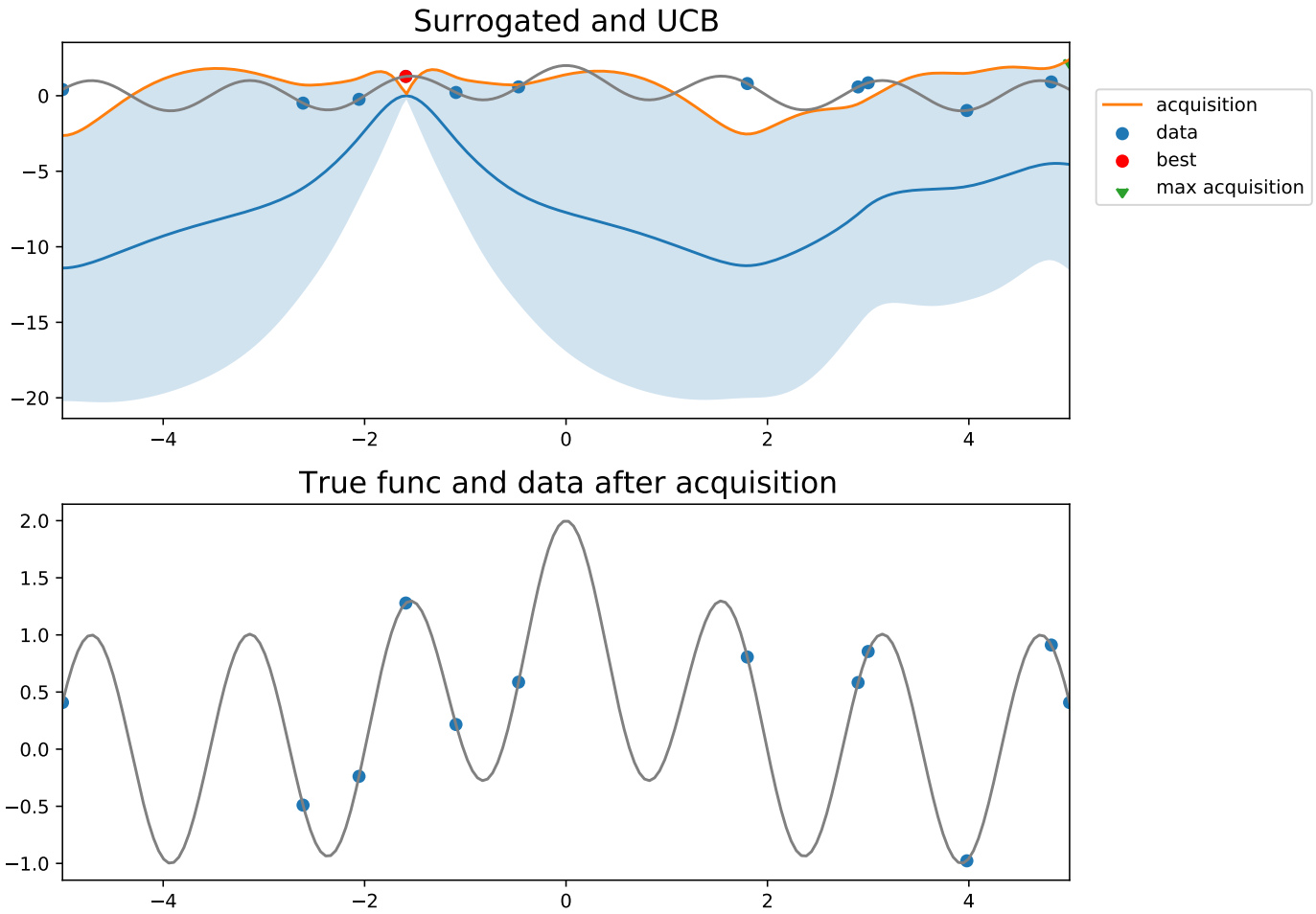
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
True func and data after acquisition

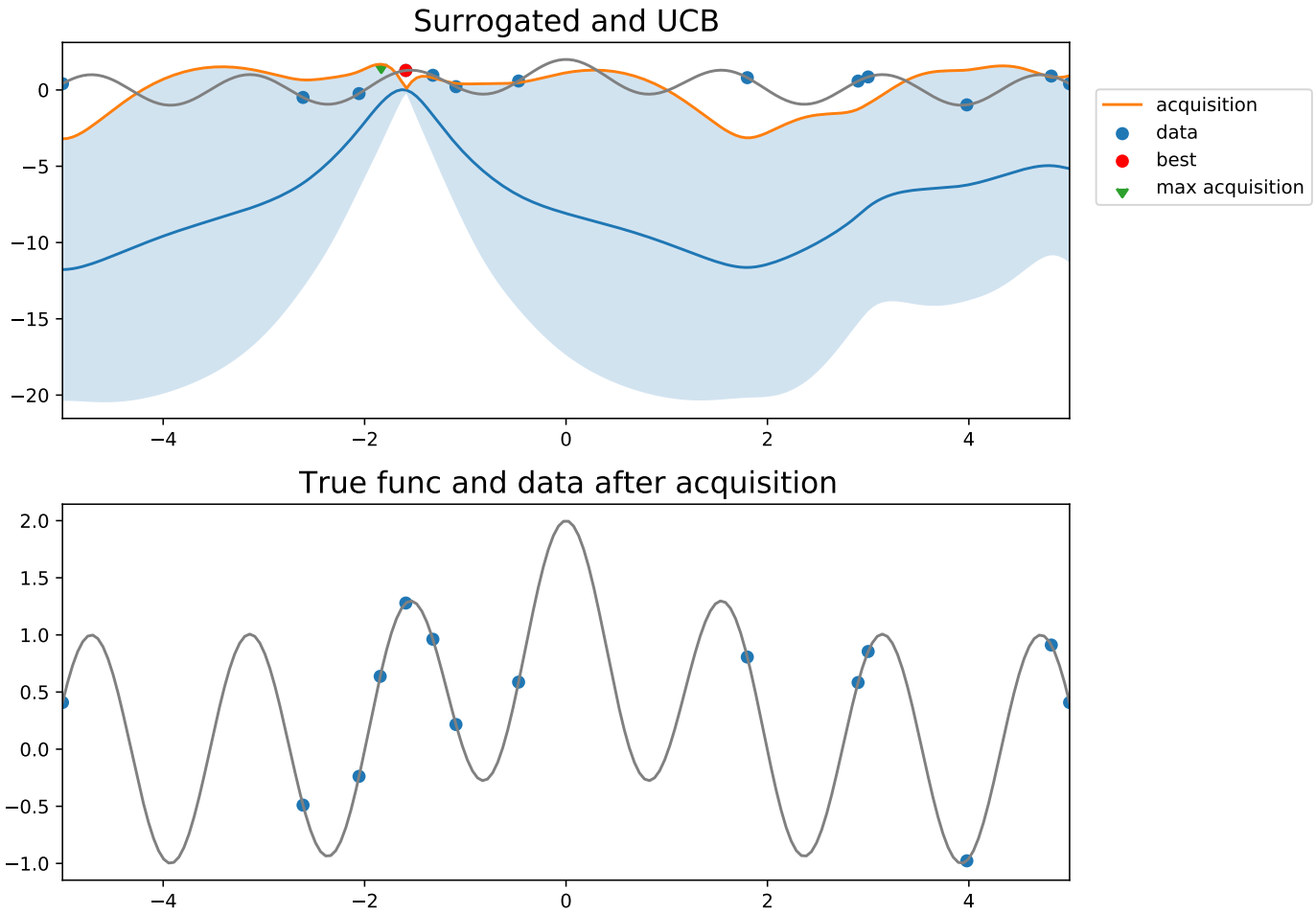
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
0
−5
−10
−15
−20
−4
−2
0
2
4
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

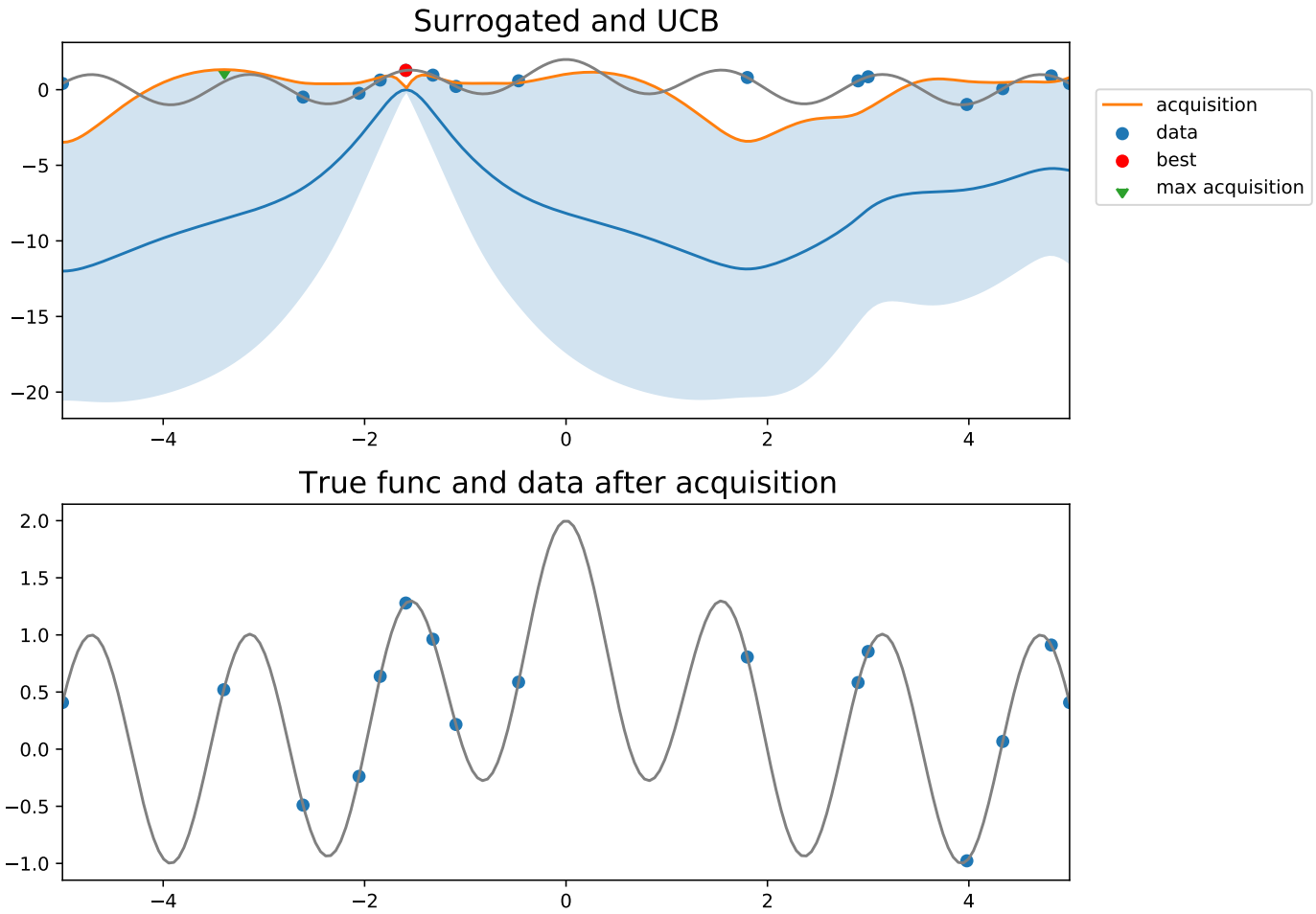
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
0
−5
−10
−15
−20
−4
−2
0
2
4
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

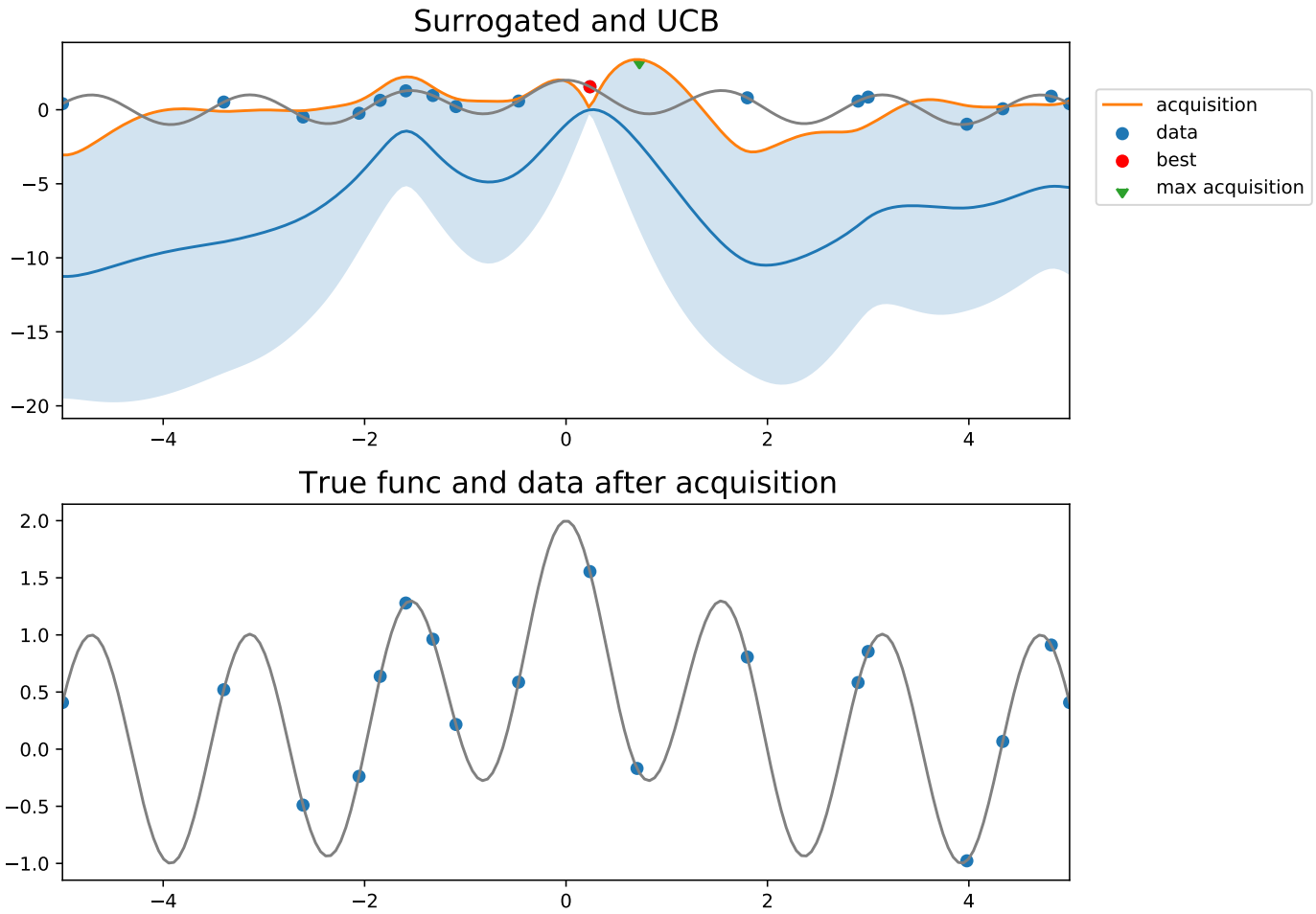
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
0
−5
−10
−15
−20
−4
−2
0
2
4
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

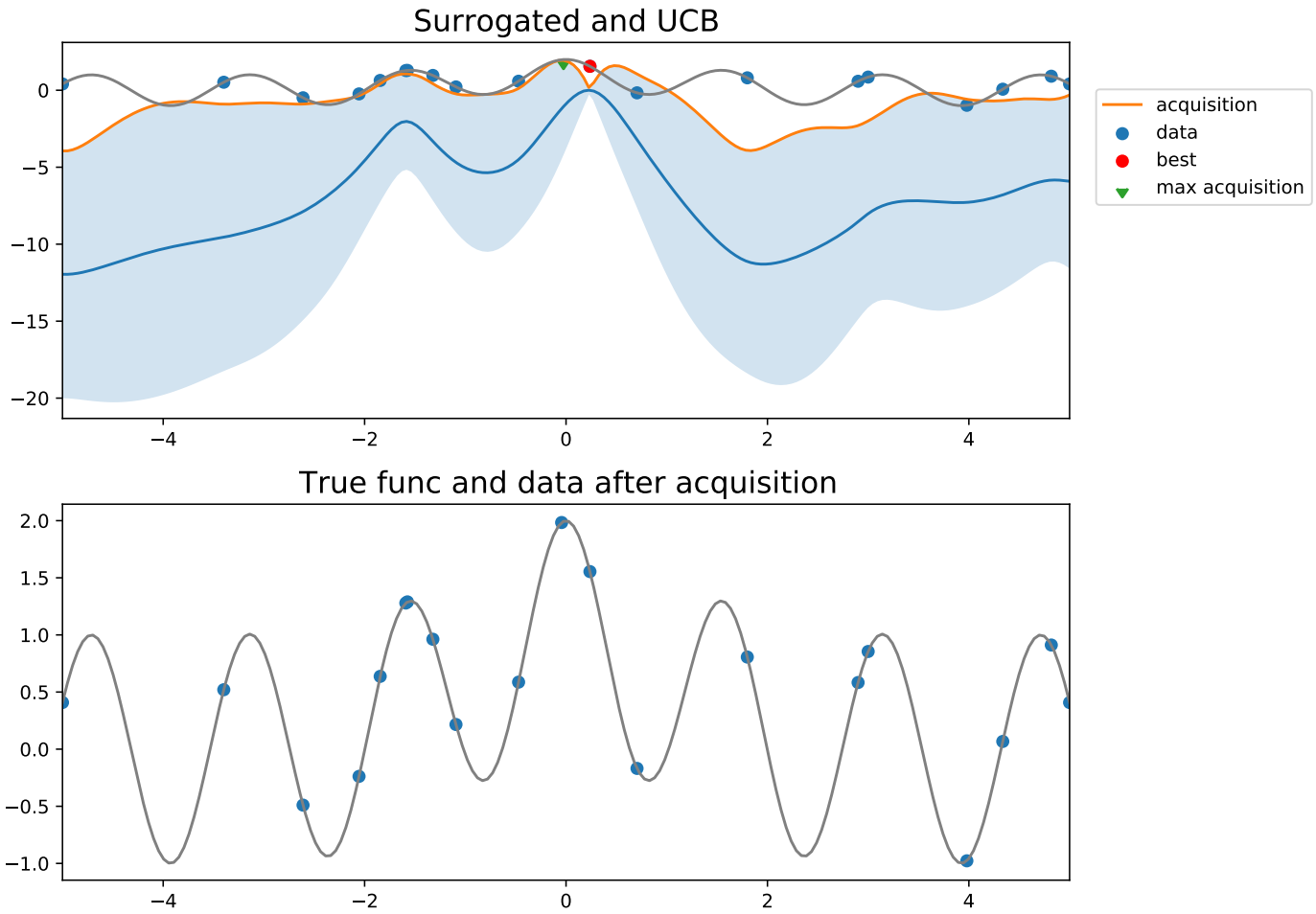
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
0
−5
−10
−15
−20
−4
−2
0
2
4
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

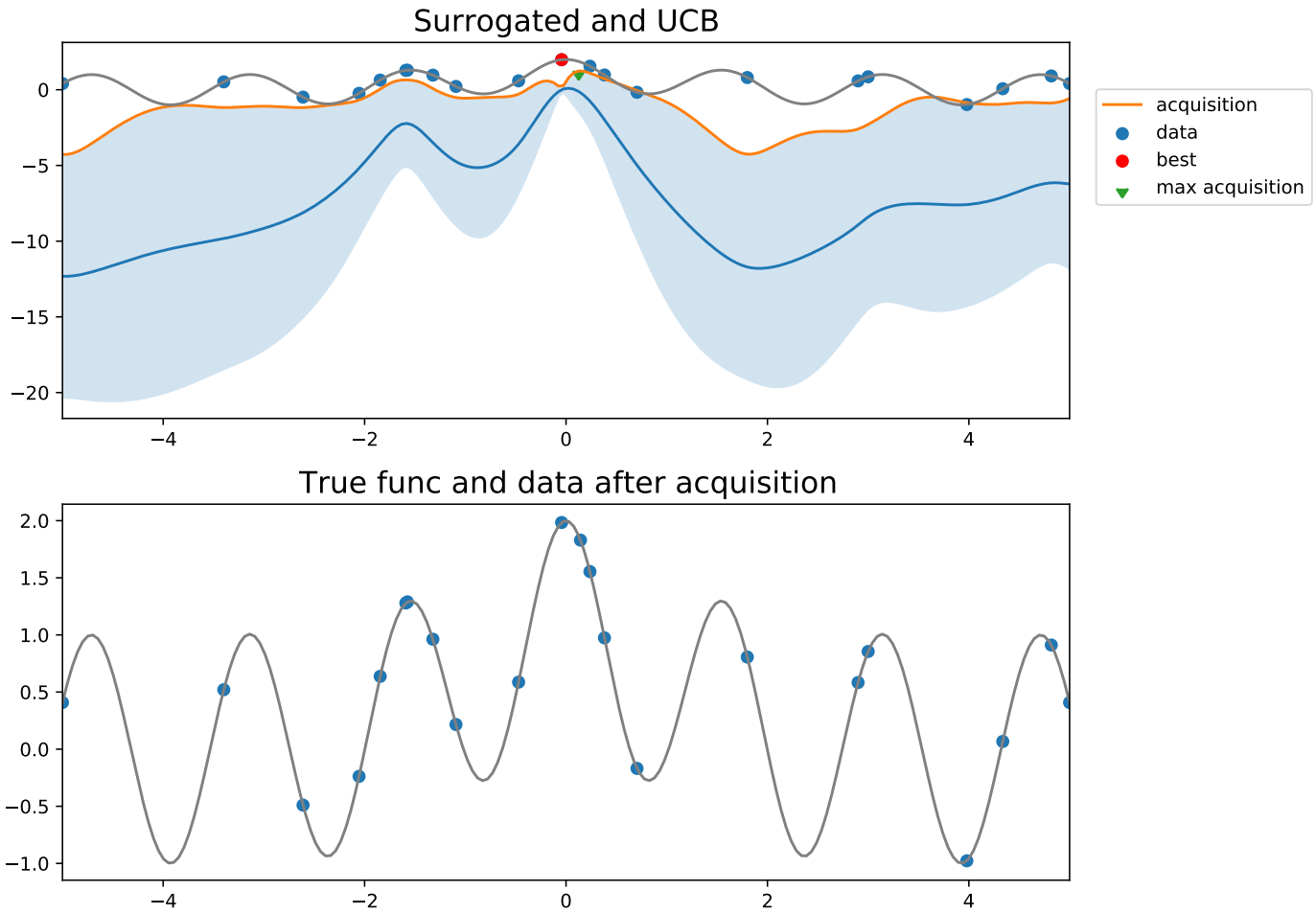
BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
0
−5
−10
−15
−20
−4
−2
0
2
4
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

BO loop


Surrogated and UCB
acquisition
data
best
max acquisition
0
−5
−10
−15
−20
−4
−2
0
2
4
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

BO loop


Surrogated and UCB
0
−5
−10
−15
−20
−4
−2
0
2
4
acquisition
data
best
max acquisition
True func and data after acquisition
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4

Batch

Manufacturing process:

It is often more convenient to work in a batch way. We prepare $n = 5$ coffees and then judge them

Judgement:

TOTAL ORDER

Likelihood

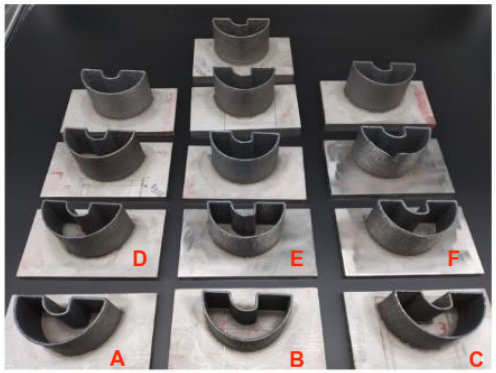
The comparisons are

$$\begin{cases} x_B \succ x_A & \text{if } f(x_B) + \text{noise}_1 > f(x_A) + \text{noise}_2 \\ x_B \succ x_C & \text{if } f(x_B) + \text{noise}_1 > f(x_C) + \text{noise}_3 \\ x_B \succ x_D & \text{if } f(x_B) + \text{noise}_1 > f(x_D) + \text{noise}_4 \\ x_B \succ x_E & \text{if } f(x_B) + \text{noise}_1 > f(x_E) + \text{noise}_5 \end{cases}$$

Assuming independent Gaussian noise $N(0, \sigma^2)$, the likelihood is:

$$p(data|f) = \int N(v_i; 0, \sigma^2) \prod_{j \in J_k} \Phi\left(\frac{f(\mathbf{x}_i) - f(\mathbf{x}_j) + v_i}{\sigma}\right) dv_i.$$

Additive Manufacturing



The process can be represented as

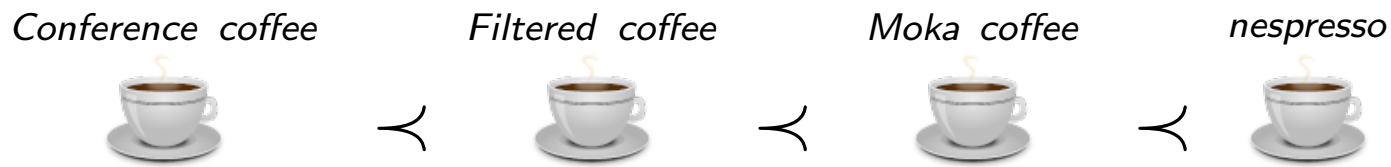
$$\text{roughness, geometry} = \mathbf{f}(\text{laser power}, \text{scan speed}, \ldots) + \textbf{noise}$$

and the optimisation problem is multi-objective.

Good: Preference based optimisation may still work if the subject is able to blend the objectives (in their head).

Bad: If we force the subject to express a preference, they can provide contradictory statements, due to the presence of competing objectives, example $A > B, \quad B > C, \quad C > A$. If we don't force the subject, they may judge them to be incomparable.

Expressing preferences is easier at the beginning



after that it becomes more difficult because ...

More difficult

Manufacturing process:

The quality of a coffee depends on many factors and this is also a multi-objective optimisation problem.

Example:

Issue:

and are bad

NO TOTAL ORDER

Generalising preferences: choice functions

Individuals are often confronted with the situation of choosing between several options (alternatives). These alternatives can be goods that are going to be purchased, candidates in elections, food etc.

We model options, that an agent has to choose, as real-valued vectors $\mathbf{x} \in \mathbb{R}^{n_x}$ and identify the sets of options as finite subsets of $\mathbb{R}^{n_x}$. Let $\mathcal{Q}$ denote the set of all such finite subsets of $\mathbb{R}^{n_x}$.

Definition

A choice function C is a set-valued operator on sets of options. More precisely, it is a map $C : \mathcal{Q} \to \mathcal{Q}$ such that, for any set of options $\mathcal{A} \in \mathcal{Q}$, the corresponding value of C is a subset $C(\mathcal{A})$ of $\mathcal{A}$.

Interpretation of choice function

$$\mathcal{A} = \{ A, B, C, D, E \}$$

$$C(\mathcal{A}) = \{ A, B, C \} \quad R(\mathcal{A}) = \{ D, E \}$$

For a given option set $\mathcal{A} \in \mathcal{Q}$, the statement that an option $\mathbf{x}_j \in \mathcal{A}$ is rejected from $\mathcal{A}$ (that is, $\mathbf{x}_j \notin C(\mathcal{A})$) means that there is at least one option $\mathbf{x}_i \in \mathcal{A}$ that an agent strictly prefers over $\mathbf{x}_j$.

Therefore choice functions represent non-binary choice models, so they are more general than preferences.

It is important to stress again that the statement $\mathbf{x}_j \notin C(\mathcal{A})$ implies there is at least one option $\mathbf{x}_i \in \mathcal{A}$ that an agent strictly prefers over $\mathbf{x}_j$. However, the agent is not required to tell us which option(s) in $C(\mathcal{A})$ they strictly prefer to $\mathbf{x}_j$.

Learning choice functions

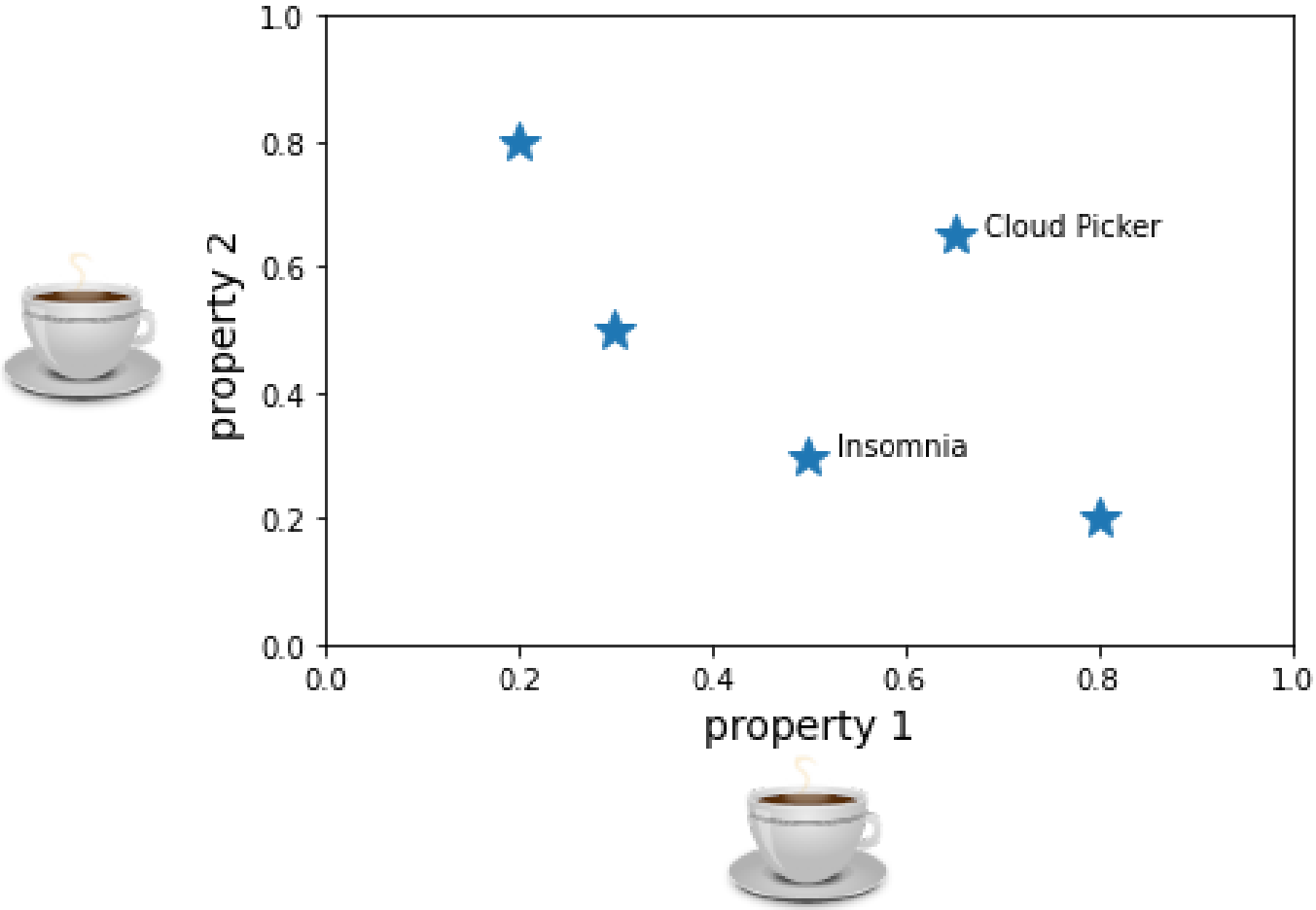
$C(\mathcal{A}) = \{$  (A),  (B),  (C) $\}$ $R(\mathcal{A}) = \{$  (D),  (E) $\}$

By following this interpretation, the set $C(\mathcal{A})$ can also be seen as the *non-dominated set* in the Pareto sense for some latent function.

Let us assume that there is a latent vector function $\mathbf{f}(\mathbf{x}_i) = [f_1(\mathbf{x}_i), \ldots, f_{n_e}(\mathbf{x}_i)]^\top$, for some dimension n_e, which embeds the options $\mathbf{x}_i$ into a space $\mathbb{R}^{n_e}$.

The choice set can then be represented through a Pareto set of non-dominated options.

Pareto non-dominated set



as higher as better

Pareto constraints

The choice of the agent expressed via $C(\mathcal{A}_k)$ implies that:

$$\neg \left(\min_{d \in D} (f_d(\mathbf{x}_i) - f_d(\mathbf{x}_j)) < 0, \quad \forall i \in I_k \right), \forall j \in J_k, \tag{1}$$

$$\min_{d \in D} (f_d(\mathbf{x}_p) - f_d(\mathbf{x}_i)) < 0, \quad \forall i, p \in I_k, p \neq i. \tag{2}$$

Condition (1) means that, for each option $x_j \in J_k$, it's not true ($\neg$ stands for logical negation) that all options in I_k are worse than x_j, i.e. there is at least an option in I_k which is better than x_j. Condition (2) means that, for each option in I_k, there is no better option in I_k. This requires that the latent functions values of the options should be consistent with the choice function implied relations. Given $\mathcal{A}_k, C(\mathcal{A}_k)$, the likelihood function $p(C(\mathcal{A}_k), \mathcal{A}_k | \mathbf{f})$ is one when (1)-(2) hold and zero otherwise.

Probabilistic model

If we include Gaussian noise, then the likelihood becomes:

$$p(C(\mathcal{A}_k), \mathcal{A}_k|\mathbf{f}) = \prod_{j \in J_k} \left(1 - \int \prod_{i \in I_k} \left(1 - \prod_{d \in D} \Phi\left(\frac{f_d(\mathbf{x}_i) - f_d(\mathbf{x}_j) - v_{dj}}{\sigma}\right)\right) N(\mathbf{v}_j; 0, \sigma^2 \mathbb{I}_d) d\mathbf{v}_j\right)$$
$$\prod_{i, p \in I_k, p \neq i} \left(1 - \prod_{d \in D} \Phi\left(\frac{f_d(\mathbf{x}_p) - f_d(\mathbf{x}_j)}{\sqrt{2}\sigma}\right)\right).$$

and we assume the prior

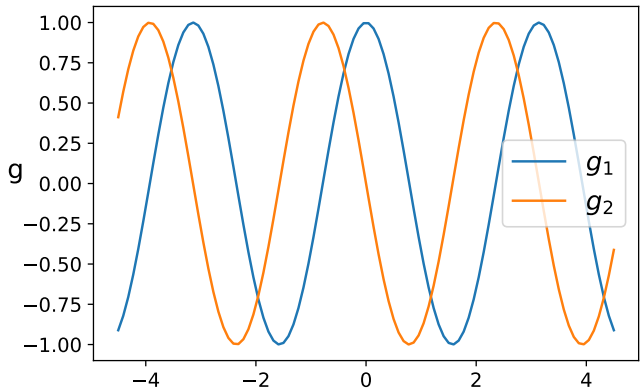
$$p(f_d) = GP(0, K), \forall d = 1, 2, \ldots, n_e$$

All the integrals in the likelihood are 1D and, therefore, can be efficiently computed using Gaussian quadrature.

Posterior: Variational + ESS

Example

Consider the 2D vector function $\mathbf{g}(x) = [\cos(2x), -sin(2x)]^\top$ with $x \in \mathbb{R}$.



We use $\mathbf{g}$ to simulate a choice function. For instance, consider the set of options $\mathcal{A}_1 = \{-1, 0, 2\}$, given that

$$\mathbf{g}(-1) = [-0.416, -0.909]$$
$$\mathbf{g}(0) = [1, 0]$$
$$\mathbf{g}(2) = [-0.65, 0.75]$$

we have that $C(\mathcal{A}_1) = \{0, 2\}$ and $R(\mathcal{A}_1) = \mathcal{A}_1 \backslash C(\mathcal{A}_1) = \{-1\}$.

Example

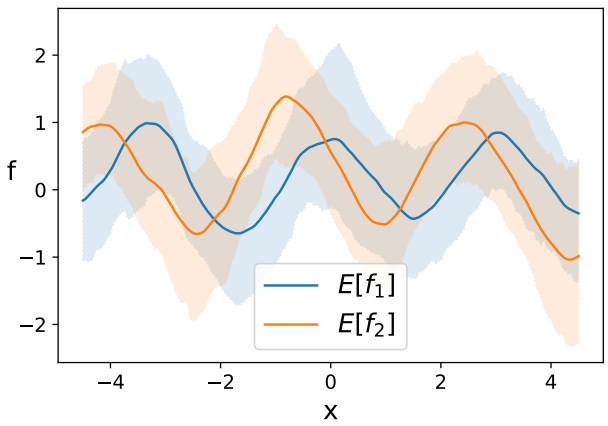
We sample 200 inputs x_i at random in $[-4.5, 4.5]$ and, using the previous approach, we generate

- $N = 50$ random subsets $\{\mathcal{A}_k\}_{k=1}^N$ of the 200 points each one of size $|\mathcal{A}_k| = 3$ (respectively $|\mathcal{A}_k| = 5$) and computed the corresponding choice pairs $(C(\mathcal{A}_k), \mathcal{A}_k)$ based on $\mathbf{g}$;
- $N = 150$ random subsets $\{\mathcal{A}_k\}_{k=1}^N$ each one of size $|\mathcal{A}_k| = 3$ (respectively $|\mathcal{A}_k| = 5$) and computed the corresponding choice pairs $(C(\mathcal{A}_k), \mathcal{A}_k)$ based on $\mathbf{g}$;

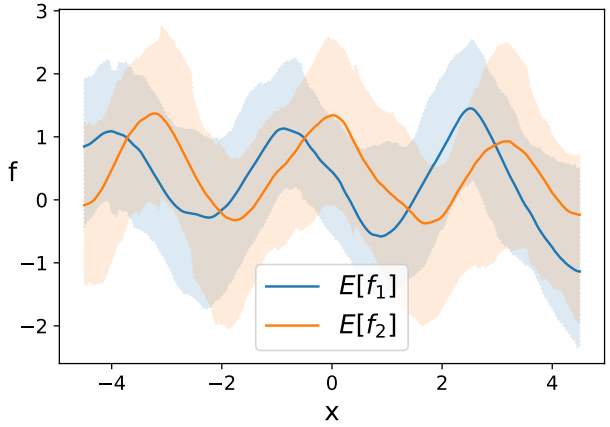
for a total of four different datasets

Example

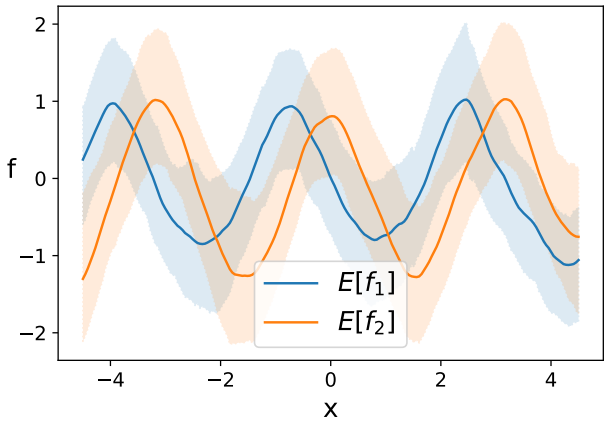
Fixing the latent dimension $n_e = 2$, we then compute the posterior means and 95% credible intervals of the latent functions:



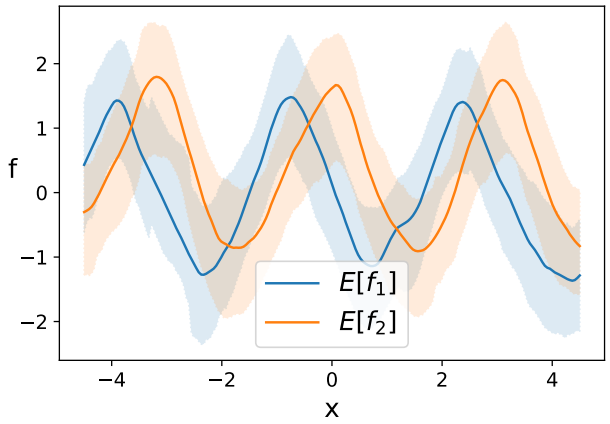
$N = 50,\ |\mathcal{A}_k| = 3$



$N = 50,\ |\mathcal{A}_k| = 5$

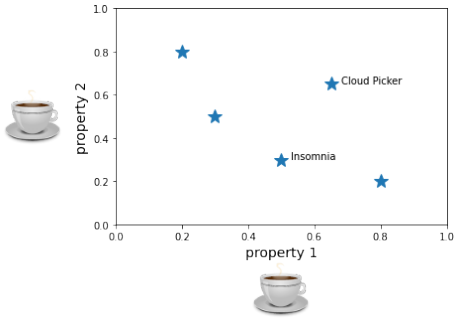


$N = 150,\ |\mathcal{A}_k| = 3$



$N = 150,\ |\mathcal{A}_k| = 5$

How do we learn the latent dimension?



The model $\mathcal{M}_{n_e}$ is conditional on the pre-defined latent dimension n_e. Although, it is sometimes reasonable to assume the number of criteria defining the choice function to be known (and so the dimension n_e), it is crucial to develop a statistical method to select n_e.

Forward selection method: we start learning the model $\mathcal{M}_1$ and we increase the dimension n_e in a stepwise manner (so learning $\mathcal{M}_2, \mathcal{M}_3, \dots$) until some model selection criterion is optimised.

Criterion: Pareto Smoothed Importance sampling Leave-One-Out cross-validation (PSIS-LOO). PSIS-loo can be computed using the samples from the posterior.

PSIS-loo

$$\varphi = \frac{1}{N}\sum_{k=1}^{N} p(z_k|z_{-k}), \tag{3}$$

where $z_k = (C(A_k), A_k)$, $z_{-k} = \{(C(A_i), A_i)\}_{i=1, i\neq k}^{N}$,

$$p(z_k|z_{-k}) = \int p(z_k|\mathbf{f})p(\mathbf{f}|z_{-k})d\mathbf{f}. \tag{4}$$

We first define the importance weights:

$$w_k^{(s)} = \frac{1}{p(z_k|\mathbf{f}^{(s)})} \propto \frac{p(\mathbf{f}^{(s)}|z_{-k})}{p(\mathbf{f}^{(s)}|\{z_k, z_{-k}\})}$$

and then approximate (4) as:

$$p(z_k|z_{-k}) \approx \frac{\sum_{s=1}^{S} w_k^{(s)} p(z_k|\mathbf{f}^{(s)})}{\sum_{s=1}^{S} w_k^{(s)}}. \tag{5}$$

PSIS-loo test 1

Generate $N = 30$ and, respectively, $N = 50$ choice data with $|\mathcal{A}_k| = 3$ from $g(x) = \cos(2x)$. 10-fold CV results.

	N=30		**N=300**	
n_e	PSIS-LOO	acc. test	PSIS-LOO	acc. test
1 (10/10)	-10	0.75	-75	0.93
2	-35	0.64	-165	0.91
3	-44	0.64	-333	0.86
4	-69	0.62	-388	0.84

PSIS-loo test 2

$\mathbf{g}(x) = [\cos(2x), -sin(2x)]^\top$ and consider three different sizes for the training dataset $N = 30, 50, 300$.

	N=30		**N=50**		**N=300**	
n_e	PSIS-LOO	acc. test	PSIS-LOO	acc. test	PSIS-LOO	acc. test
1	-56	0.20	-89	0.23	-493	0.30
2	-39	0.32	-47	0.51	-236	0.72
3	-39	0.32	-49	0.49	-269	0.65
4	-42	0.30	-53	0.43	-277	0.64

More difficult

Manufacturing process:

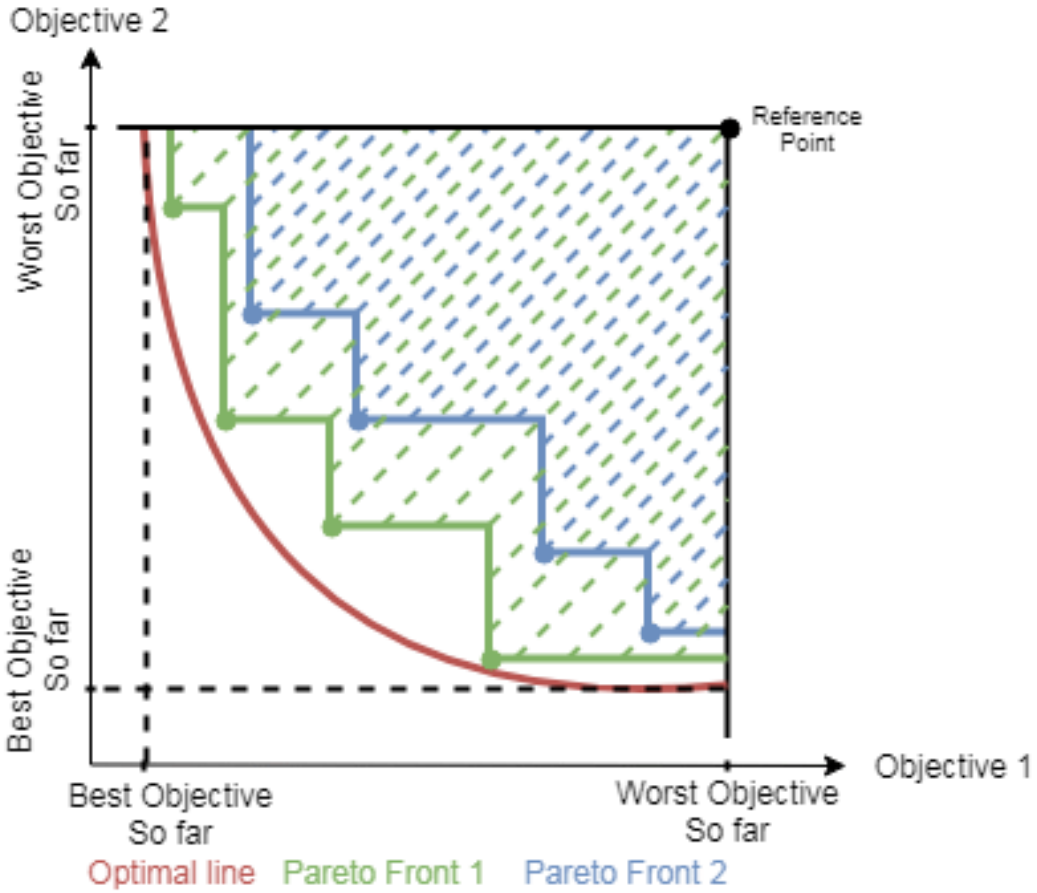
The quality of a coffee depends on many factors and this is also a multi-objective optimisation problem.

Example:

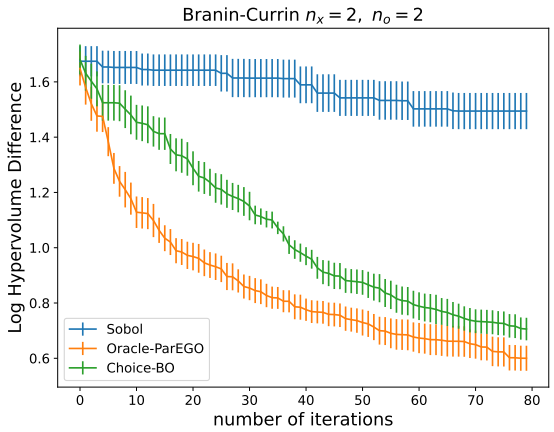
Issue:

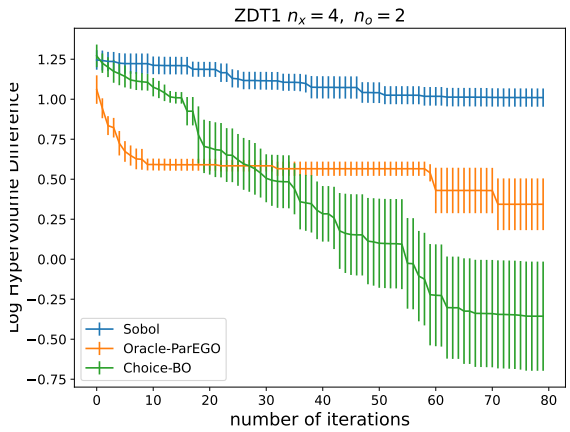
NO TOTAL ORDER

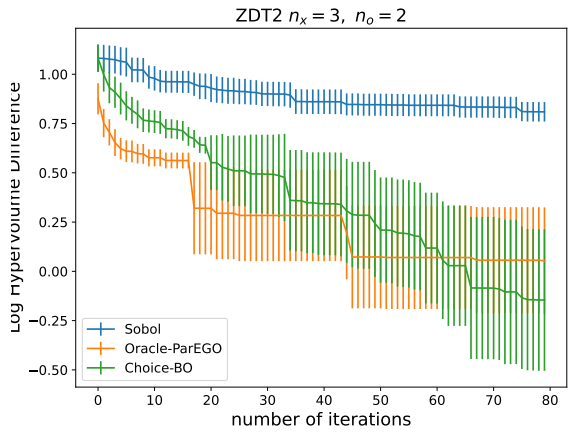
Multi-objective BO

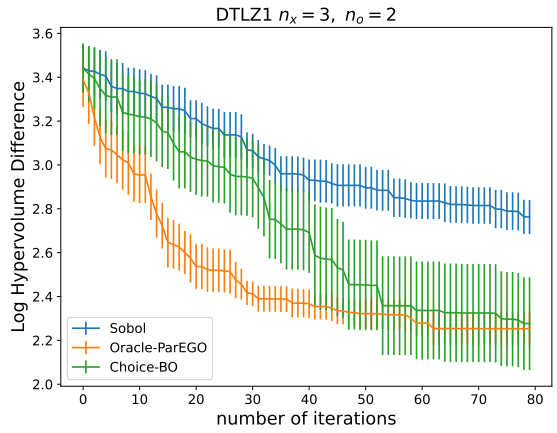


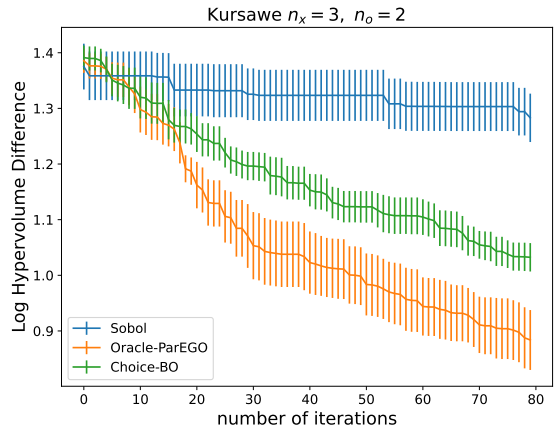
Multi-objective BO

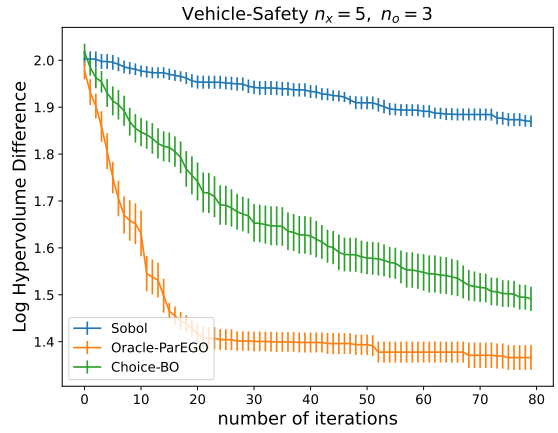












Conclusions

- Choice functions to express judgements;
- A Bayesian model to learn from choice data (including learning the latent dimension n_e)
- Application to multi-objective Bayesian BO;

Future work:

- Each BO iteration needs between 30s (20 choices) to 180s (100 choices). It is necessary to find a way to speed up the process.