Bayesian optimisation using Stan

Anna Elisabeth Riha   Adam Howes   Seth Flaxman   Elizaveta Semenova

September 13, 2024

Bayesian optimisation using Stan

(a nice BO image here)

In this tutorial, we will

- provide an overview of Bayesian optimisation (BO),
- demonstrate how Stan can be used within the BO procedure,
- demonstrate variable query cost and non-response propensity within BO.

Learning outcomes for today's tutorial

After this session, you will be able to

- formulate the main goal and components of BO,
- implement Gaussian process surrogates,
- use several acquisution functions within BO.

Schedule for today's tutorial

Time	Activity
5 min	Warmup
10 min	What does Bayesian optimisation solve?
10 min	Building blocks of Bayesian optimisation
20 min	Surrogates
10 min	Acquisition functions
10 min	Break
5 min	Computational tricks
10 min	Cost- and propensity-aware BO
20 min	Hands-on practice

Warmup

Where to sample next to find global minimum?

TODO

What does Bayesian optimisation solve?

Problem definition: global optimisation

The goal of **global optimisation** of a real-valued function $f : \mathcal{X} \to \mathbb{R}$ is to find a *minimiser* x^* (there may be more than one) in the search space $\mathcal{X}$, such that:

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

Problem definition: global optimisation

Today we focus on finding a minimum, but finding a maximum can be approached in the same way since

$$\max_{x \in \mathcal{X}} f(x) = -\min_{x \in \mathcal{X}} f(x).$$

Problem definition: global optimisation

The function f to be optimised is referred to as the **objective function**.

In contrast to *local optimisation*, global optimisation requires that

$$f(x^*) \leq f(x)$$

for **all** $x \in \mathcal{X}$ rather than only in some neighbourhood about x^*. Throughout this workshop, we assume that the search space $\mathcal{X}$ is a subset of $\mathbb{R}^d$ where $d \in \mathbb{N}$:

$$\mathcal{X} \subset \mathbb{R}^d$$

Group discussion

Given a function $f : \mathcal{X} \to \mathbb{R}$, how would you approach the search of its minimum?

Problem definition: global optimisation

In practice, the objective function f may possess the following challenging properties:

1. *Non-linear, non-convex*,

Problem definition: global optimisation

In practice, the objective function f may possess the following challenging properties:

2. *Black-box:* A function is called **black-box** if it can only be viewed in terms of its inputs and outputs. If f is black-box then it does not have an analytic form or derivatives, such as the gradient ∇f or Hessian $\mathbf{H}$.

Problem definition: global optimisation

"Any sufficiently complex system acts as a black-box when it becomes easier to experiment with than to understand."

Golovin et al, "Google Vizier" (2017)

Problem definition: global optimisation

In practice, the objective function f may possess the following challenging properties:

3. *Expensive to evaluate*: The sampling procedure is computationally, economically or otherwise prohibitively expensive.

Problem definition: global optimisation

In practice, the objective function f may possess the following challenging properties:

4. *Noisy*: When f is evaluated at x, the value returned y is contaminated by noise ϵ, typically assumed to be Gaussian with zero mean and variance σ^2 such that

$$y = f(x) + \epsilon$$

.

Problem definition: global optimisation

- Perhaps, the global optimisation problem can be solved using sampling!

Sample designs

- How should points be queried to efficiently learn about x^*?

Sample designs

- How should points be queried to efficiently learn about x^*?
- Let's focus on finding a "good" solution or converging to a minimiser x^* in few evaluations, rather than in making theoretical guarantees about optimality.

Sample designs

The two relatively naive strategies:

- *grid-search*,
- *random-search*.

Sample designs: grid search

Grid-search:

- **How**: Samples are taken spaced evenly throughout the domain $\mathcal{X}$ at a resolution appropriate to the optimisation budget.
- **Pitfalls**: Although the whole domain is superficially covered, if few function evaluations can be afforded then this coverage is too sparse to reliably locate a minimiser.

Sample designs: random search

Random-search:

- **How**: random-search chooses inputs in the domain $\mathcal{X}$ to evaluate at random.
- **Pitfalls**: complete randomness lends itself to clumps of points and large areas left empty.

Sample designs: latin-hypercube

Latin-hypercube:

a grid with exactly one sample in each column and each row. This avoids the problem of collapsing, from which grid-search suffers.

Sample designs: static designs

An issue with the aforementioned strategies: information gained during the search is not used to better inform the next decision.

Sample designs: sequential decision making

Rather than choose all the points at once it makes sense instead to consider a *sequential decision making* problem where at each stage a point is carefully chosen to be evaluated next.

Sample designs: sequential decision making

So, how can previous information be used?

Sequential decision making: idea

- Make assumptions about f, e.g. that f is smooth.

Sequential decision making: idea

- Make assumptions about f, e.g. that f is smooth.
- Build a model which mimics behaviour of f. A model with uncertainty!

Sequential decision making: idea

- Make assumptions about f, e.g. that f is smooth.
- Build a model which mimics behaviour of f. A model with uncertainty!
- Sample in uncertain regions, not near to any previous evaluations, to *explore*.

Sequential decision making: idea

- Make assumptions about f, e.g. that f is smooth.
- Build a model which mimics behaviour of f. A model with uncertainty!
- Sample in uncertain regions, not near to any previous evaluations, to *explore*.
- Sample in promising regions, near to previous low evaluations, to *exploit*.

Sequential decision making: idea

This is exactly how Bayesian optimisation works.

Buildling blocks of Bayesian optimisation

What is bayesian optimisation?

So, what is Bayesian optimisation exactly?

What is bayesian optimisation?

- **Bayesian optimisation** (BO) (Mockus, Tiesis, and Zilinskas (1978)) is a statistical / machine learning technique for addressing the global optimisation task by treating the objective function $f(.)$ as a *random variable*.

What is bayesian optimisation?

- **Bayesian optimisation** (BO) (Mockus, Tiesis, and Zilinskas (1978)) is a statistical / machine learning technique for addressing the global optimisation task by treating the objective function $f(.)$ as a *random variable*.
- It enables the optimisation of “black-box” functions.

What is bayesian optimisation?

- **Bayesian optimisation** (BO) (Mockus, Tiesis, and Zilinskas (1978)) is a statistical / machine learning technique for addressing the global optimisation task by treating the objective function $f(.)$ as a *random variable*.
- It enables the optimisation of “black-box” functions.
- It is a *sequential*, *model-based* optimisation algorithm which learns from all available observations to make better informed choices about the next point to query.

BO components: surrogates

We need a model for the objectve “black-box” function. Where to get it?

BO components: surrogates

- Since f is black-box, there is uncertainty about it its values:

BO components: surrogates

- Since f is black-box, there is uncertainty about it its values:
    - where it has not been directly evaluated,

BO components: surrogates

- Since f is black-box, there is uncertainty about it its values:
  - where it has not been directly evaluated,
  - when f is noisy, then even at evaluated points as well.

BO components: surrogates

A perfect case for Bayesian inference and Stan to do it for us!

BO components: surrogates

- We can build an explicit probabilistic model of f, known as a **surrogate** (or **emulator**) $g_\theta(x)$.

BO components: surrogates

- We can build an explicit probabilistic model of f, known as a **surrogate** (or **emulator**) $g_\theta(x)$.
- The emulator provides a *predictive distribution* for f evaluated at new inputs.

BO components: surrogates

- We can build an explicit probabilistic model of f, known as a **surrogate** (or **emulator**) $g_\theta(x)$.
- The emulator provides a *predictive distribution* for f evaluated at new inputs.
- When data $(x_i, y_i), i = 1, ..., t$ is observed then the model can be sequentially updated and refined using the Bayes' rule.

BO components: surrogates

The surrogate $g_\theta(x)$ allows to replace the task of global optimisation of the objective function $f(.)$ with the task of global optimisation of the surrogate $g_\theta(.)$:

$$\arg\min_{x \in \mathcal{X}} f(x) \approx \arg\min_{x \in \mathcal{X}} g_\theta(x).$$

BO components: surrogates

Requirements to the surrogate:

- It should behave similarly to the objective function f,
- It should be cheaper to evaluate at a point x than the target function $f(x)$, and allow to quantify uncertainty of how well $g_\theta(x)$ approximates $f(x)$

BO components: surrogates

Based on noisy observations

$$y_i = f(x_i) + \epsilon_i$$

and collected are observation pairs (x_i, y_i) we fit the surrogate to the data

$$g_\theta(x_i) \approx y_i.$$

Acquisition functions

We have obtained a predictive distribution using $g_\theta(x)$. Now what?

Acquisition functions

At every iteration, an **acquisition function** $a(x)$ is used to decide which point x_{t+1} to query next as x_{t+1} is chosen as the optimum of $a(x)$:

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} a(x)$$

.

BO components: summary

In summary, BO is a sequential *model-based* optimisation algorithm which learns from all available observations to make better informed choices about the next point to query, and uses a probabilistic surrogate model representing the objective function for it.

The BO loop

The Bayesian optimisation procedure

- *Initialization*: select initial points (x_0, y_0) to evaluate the objective function
- *Iterative Loop*:
  - update the **surrogate model** by fitting $g_\theta(x_i) \approx y_i$, $i = 0, ..., t$.
  - evaluate **acquisition function** $a(x)$
  - select the next point as the optimnum of the acquisition function to $x_{t+1} = \arg\min_{x \in \mathcal{X}} a(x)$.
  - evaluate the objective function: $y_{t+1} = f(x_{t+1}) + \epsilon_i$
  - Stopping Criterion: check if the stopping criterion is met (e.g., maximum number of iterations reached, convergence achieved, or budget exhausted).
- *Termination*: Return the best observed point or the point that minimises the surrogate model's prediction of the objective function.

Bayesian optimisation use cases

- Machine learning
- Drug development
- Chemical discovery
- Climate models
- Robotic control

Connection to active learning

- Active learning, on the other hand, focuses on selecting the most informative data points in order to learn the whole function.
- Put simply, BO is active learning for global optimisation.

Let’s take a break! (10 min)

Some suggestions for recharging during breaks :)

- move your body
- open a window or go outside
- drink some water
- try to avoid checking e-mails, messengers, or social media

Gaussian Processes as surrogates

$$p(\theta \mid y) \propto p(\theta)p(y \mid \theta)$$

XYZ

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Summary

What we learnt today:

- What is BO

Summary

What we learnt today:

- What is BO
- BO components: surrogates and acquisition functions

Summary

What we learnt today:

- What is BO
- BO components: surrogates and acquisition functions
- Cost- and response propensity-aware BO

More resources

Reading materials

- a paper on . . . :
- a paper on . . . :

Case studies

- a case study on . . . : add url
- a case study on . . . : add url

Stay tuned

We will soon release a write-up based on the materials of this workshop, with more details.

References

Kushner, Harold J. 1964. "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise."

Mockus, Jonas, Vytautas Tiesis, and Antanas Zilinskas. 1978. "The Application of Bayesian Methods for Seeking the Extremum." *Towards Global Optimization* 2 (117-129): 2.