Introduction to Bayesian Optimization

and more generally to the Design & Analysis of Computer Experiments (DACE)

Julien Bect

Université Paris-Saclay, CNRS, CentraleSupélec
Laboratoire des Signaux et Systèmes (L2S), Gif-sur-Yvette, France
`https://l2s.centralesupelec.fr/`

GdR MASCOT-NUM, INSMI, CNRS
`https://www.gdr-mascotnum.fr/`

15th Peyresq Summer School
in Signal and Image Processing
Online, 2021, June 20–26

What is Bayesian optimization ?

- "wide sense" definition
  - optimization using tools from Bayesian UQ

What is Bayesian optimization ?

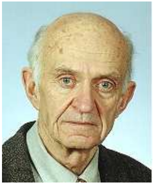
- “wide sense” definition
  - optimization using tools from Bayesian UQ
  - started with Harold Kushner’s paper: *A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise*, J. Basic Engineering, 1964.

What is Bayesian optimization ?

- a slightly more restrictive definition
  - sequential Bayesian decision theory applied to optimization

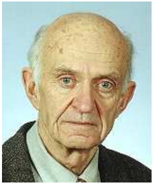
What is Bayesian optimization ?

- a slightly more restrictive definition
  - sequential Bayesian decision theory applied to optimization
  - started with the work of Jonas Mockus and Antanas Žilinskas in the 70's, e.g., *On a Bayes method for seeking an extremum*, Avtomatika i Vychislitel'naya Teknika, 1972 (in Russian)



What is Bayesian optimization ?

- a slightly more restrictive definition
  - sequential Bayesian decision theory applied to optimization
  - started with the work of Jonas Mockus and Antanas Žilinskas in the 70's, e.g., *On a Bayes method for seeking an extremum*, Avtomatika i Vychislitel'naya Teknika, 1972 (in Russian)



- In this lecture we adopt this second (more constructive !) definition

Decision-theoretic framework

Decision-theoretic framework

- Bayesian decision theory (BDT) in a nutshell
  - a mathematical framework for decisions under uncertainty
  - uncertainty is captured by probability distributions
  - the "Bayesian agent" aims at minimizing the expected loss

Decision-theoretic framework (cont'd)

- How does this relate to optimization ?

(In a general BDT problem, the Bayesian agent itself can also have a state, that changes as a consequence of the decisions; think, e.g., of a robot planning problem: the state could be the position & energy status of the robot.)

Decision-theoretic framework (cont'd)

- How does this relate to optimization ?
- The agent is the optimization algorithm (or you, if you will)

(In a general BDT problem, the Bayesian agent itself can also have a state, that changes as a consequence of the decisions; think, e.g., of a robot planning problem: the state could be the position & energy status of the robot.)

Decision-theoretic framework (cont'd)

- How does this relate to optimization ?
- The agent is the optimization algorithm (or you, if you will)

Ingredients of a BDT problem

- a set Ω of all possible "states of nature"
- a prior distribution P_0 over the states of nature
- a description of the decisions we have to make
- and the corresponding "transitions"
- a loss function L (or utility function U)

(In a general BDT problem, the Bayesian agent itself can also have a state, that changes as a consequence of the decisions; think, e.g., of a robot planning problem: the state could be the position & energy status of the robot.)

Important example: single-objective optimization

- Consider the following setting
  - a deterministic numerical model with scalar output:

$$\begin{aligned} f : \mathbb{X} &\to \mathbb{R} \\ x &\mapsto f(x) \end{aligned}$$

  - "known" input space $\mathbb{X} \subset \mathbb{R}^d$; e.g., $\mathbb{X} = [0;1]^d$
  - f assumed expensive to evaluate; gradient not available

- Optimization problem: find
  - $m^* = \min_{\mathbb{X}} f$
  - and/or $x^* = \operatorname{argmin}_{\mathbb{X}} f$

(Until further notice, we will use this simple—but important—setting to present the basics of Bayesian optimization.)

Important example: single-objective optimization (cont'd)

- States of nature:

$\Omega = \mathbb{R}^{\mathbb{X}} = \{ \text{all functions } f : \mathbb{X} \to \mathbb{R} \}$

or $\Omega = C(\mathbb{X}; \mathbb{R})$ ...

- Prior distribution:

$P_0 = GP(m, k)$

$= \int GP(m_\theta, k_\theta)\, \pi(\theta)\, d\theta$ hierarchical

Important example: single-objective optimization (cont'd)

- Intermediate decisions:

  $X_1, X_2, \ldots X_N$: evaluation points $\in \mathbb{X}$

  (alt. : batches)

- Transitions of the "state" of the Bayesian agent:

  $P_0 \to P_1 \to \cdots \to P_n \to P_{n+1} \to \cdots$

  $P_n = P_0 (\cdot \mid \mathcal{F}_n)$

Notation: $\mathcal{F}_n = (X_1, \xi(X_1), \ldots, X_n, \xi(X_n))$.

Important example: single-objective optimization (cont'd)

- Stopping decision: when to stop sampling

$$N = \underline{N_{\text{budget}}} \qquad \text{(prescribed budget)}$$

- Terminal decision: c'est votre dernier mot? (J. P. Foucault)

$$D_{N+1} = \hat{X} \quad \rightsquigarrow \quad \text{estimate of the minimizer}$$

$$= \hat{M} \quad \rightsquigarrow \quad \text{———— the minimum}$$

$$= (\hat{X}, \hat{M})$$

Important example: single-objective optimization (cont'd)

- Loss: the opportunity cost (a.k.a linear loss, L^1 loss, simple regret...)

$$d = \hat{x} \in \mathbb{X}$$

$$L(f, d) = f(\hat{x}) - \min f = |f(\hat{x}) - \min f|$$

- A more "conservative" loss:

$$d = (\hat{x}, \hat{m})$$

$$L(f, d) = \begin{cases} \hat{m} - \min f & \text{si } f(\hat{x}) = \hat{m} \\ +\infty & \text{sinon} \end{cases}$$

(If instead of point estimates we choose to provide probalistic estimates in the form of predictive density functions, then we can also consider the *negative log* loss, which leads to entropy-based methods.)

More decisions?

- Intermediate decisions: various extensions
  - parallel computing: batches of input values
  - multi-fidelity: choosing the right fidelity level
  - tunable run-time: choosing when to stop a computation
  - ...
- Stopping decision: optimal stopping?
  - stopping based on some target accuracy on x^* and/or m^*
  - trade-off between observation cost and accuracy
  - ...
- Final decision: other settings
  - multi-objective: Pareto set / Pareto front,
  - quasi-optimal region (sublevel set)
  - ...

Sequence of decision rules

- We are looking for a sequence of decision rules
  - a.k.a. policy, or strategy
  - Notation:

$$\underline{D}(f) = (X_1(f), \ldots, X_N(f), D_{N+1}(f)), \qquad f \in \Omega.$$

- We cannot use information that is not yet available
  - $X_n(f)$ depends on f through $\mathcal{F}_{n-1}$ only ($\forall n \leq N$)
  - $D_{N+1}(f)$ depends on f through $\mathcal{F}_N$ only

- Loss = terminal cost: $L(\cancel{\omega}\, f, \underline{d}) = L(\cancel{\omega}\, f, d_{N+1})$
  - where $\underline{d} = (x_1, \ldots, x_N, d_{N+1}) \in \mathbb{X}^n \times \mathbb{D}$

The Bayesian way

- Bayes-optimal strategy (optimization algorithm):

$$\begin{aligned} \underline{D}^{\text{Bayes}} &= \operatorname{argmin}_{\underline{D}} \; \mathsf{E}_0 \left(L(\xi, D_{N+1}) \right) \\ &= \operatorname{argmin}_{\underline{D}} \int_\Omega L(f, D_{N+1}(f)) \, \mathsf{P}_0(\mathrm{d}f) \end{aligned}$$

where $\underline{D}$ ranges over all strategies $\underline{D} = (X_1, \ldots, X_N, D_{N+1})$

- Problem: find $\underline{D}^{\text{Bayes}} \ldots$

Can we actually build an optimal Bayesian algorithm?

Optimal terminal decision

- Define the posterior risk at time N for the decision d_{N+1}:

$$R_N(d_{N+1}) = \mathsf{E}\left(L(\xi,\, \underline{d_{N+1}}) \mid \mathcal{F}_N\right)$$

("risk" is a synonym for "expected loss")

formule de l'espérance totale

- Then…

$$E_0\left(L(\xi, D_{N+1})\right) = E_0\left(E_0\left(L(\xi, d_{N+1}) \mid \mathcal{F}_N\right)\right)$$

$$= E_0\left(\underbrace{R_N\left(\underline{d_{N+1}}\right)}\right)$$

$$\Rightarrow \boxed{D^*_{N+1} = \operatorname{argmin}_{d_{N+1}} R_N\left(d_{N+1}\right)}$$

Example (cont'd): the modified linear loss

Consider the case where $d_{N+1} = (\widehat{x}, \widehat{m})$ and

$$L(f, d_{N+1}) = \begin{cases} \widehat{m} - \min f & \text{if } f(\widehat{x}) = \widehat{m} \\ +\infty & \text{otherwise.} \end{cases}$$

Assume a non-degenerate GP model: $\xi \mid \mathcal{F}_N \sim \mathrm{GP}(\widehat{\xi}_N, k_N)$ with

$$k_N(x,x) = 0 \quad \text{iff} \quad x \in \{X_1, \ldots, X_N\}$$

Then...

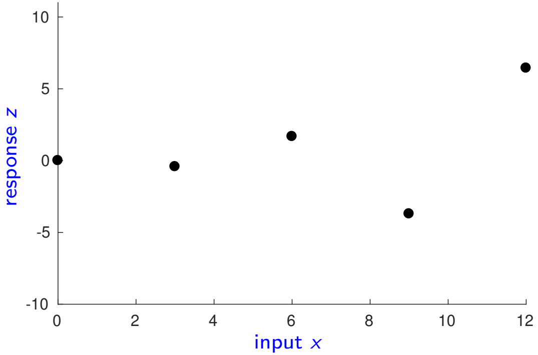
$$R_N(d_{N+1}) = \begin{cases} \hat{m} - E(\min \xi \mid \mathcal{F}_N) & \text{if } \xi(\hat{x}) = \hat{m} \;\; P_N\text{-as} \\ +\infty & \text{otherwise} \end{cases}$$

$$\Rightarrow \boxed{D^*_{N+1} = (X_{i^*}, \xi(X_{i^*}))}$$

with i^* s.t. $\xi(X_{i^*}) = \min_{i \le N} \xi(X_i)$

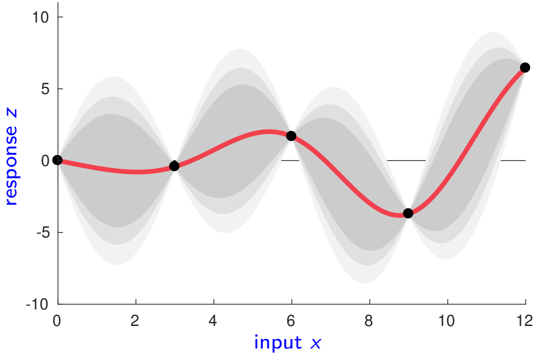
Example (cont'd): the modified linear loss

Assume that $n = N = 5$ (a small budget indeed).



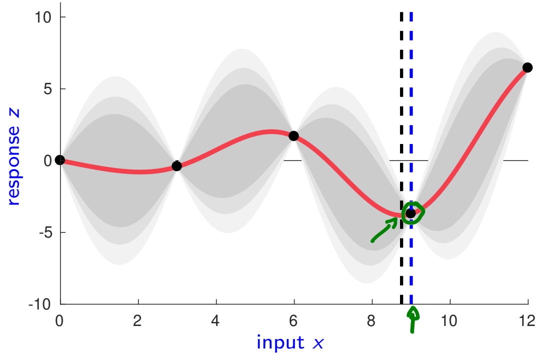
Example (cont'd): the modified linear loss

Assume that $n = N = 5$ (a small budget indeed).



Example (cont'd): the modified linear loss

Assume that $n = N = 5$ (a small budget indeed).



blue dashed line: $\widehat{X} = \operatorname{argmin}_{i \leq n} \xi(X_i)$

black dashed line: $\widehat{X} = \operatorname{argmin}_{x \in \mathbb{X}} \cancel{\widehat{X}}\ \xi_n(x_i)$ —

Optimal choice of the last evaluation point

- Bayes risk at time N:

$$R_N^* \;=\; \min_{d_{N+1}\in\mathbb{D}} R_N(d_{N+1}) \;=\; R_N(D_{N+1}^*)$$

- Posterior risk at time $N-1$:

$$R_{N-1}(x_N) \;=\; \mathsf{E}_0\left(L(\xi, D_{N+1}^*) \mid \mathcal{F}_{N-1},\, X_N = x_N\right)$$
$$= \mathsf{E}_0\left(R_N^* \mid \mathcal{F}_{N-1},\, X_N = x_N \right)$$

$$\Rightarrow \quad \mathsf{E}_0\left(L(\xi, D_{N+1}^*) \right) = \mathsf{E}_0\left(R_{N-1}(x_N) \right)$$

- Then

$$\boxed{X_N^* \;=\; \operatorname{argmin}_{x_N \in \mathbb{X}} R_{N-1}(x_N)}$$

- Remark: R_{N-1} is used as a "sampling criterion"

(a.k.a. "infill criterion", a.k.a. "merit function"...)

Example (cont'd): the modified linear loss

- Set $M_n = \min_{i \leq n} \xi(X_i)$, $n \leq N$.
- Recall: $R_N^* = M_N - \mathrm{E}\left(\min \xi \mid \mathcal{F}_N\right)$
- Then

$$R_{N-1}(x_N) = E_0\left(M_N - E(\min \xi \mid \mathcal{F}_N) \mid \mathcal{F}_{N-1}, X_N = x_N\right)$$
$$= E_0\left(M_N \mid \mathcal{F}_{N-1}, X_N = x_N\right) + \text{const}$$

$$X_N^* = \operatorname{argmin}_{x_N \in \mathbb{X}} E_0\left(\min\left(M_{N-1}, \xi(x_N)\right) \mid \mathcal{F}_{N-1}\right)$$
$$= \operatorname{argmax}_{x_N \in \mathbb{X}} E_0\left(\left(M_{N-1} - \xi(x_N)\right)_+ \mid \mathcal{F}_{N-1}\right)$$

- This is the Expected Improvement (EI) criterion

(Mockus et al 1978; Jones, Schonlau & Wlech, 1998)

(computable analytically for GP priors ⇒ very commonly used)

One-dimensional illustration

STK demo (https://github.com/stk-kriging/stk)

`stk_example_doe03`

- One-dimensional illustration
- Expected Improvement (EI) criterion
- Ordinary kriging, Matérn-5/2 covariance function (known parameters)
- For now we will only look at the final stage of this demo.

Back to the Bayes-optimal strategy

- Notation: $\mathsf{E}_{n,x} = \mathsf{E}_0\left(\,\cdot \mid \mathcal{F}_n, X_{n+1} = x\right)$.
- Backward induction (or dynamic programming):

$$X_1^* = \operatorname{argmin}_{x_1} \mathsf{E}_{0,x_1}\left(\min_{x_2} \mathsf{E}_{1,x_2}\left(\ldots\right.\right.$$

$$\underbrace{\min_{x_N}}_{x_N^*} \mathsf{E}_{\underline{N-1},x_N}\left(\underbrace{\min_d}_{\xi_\alpha\ D_{un}} \mathsf{E}_N\left(\underline{L(\xi, d)}\right)\right)\left.\left.\right)\right)$$

Back to the Bayes-optimal strategy

- Notation: $\mathsf{E}_{n,x} = \mathsf{E}_0\left(\,\cdot \mid \mathcal{F}_n, X_{n+1} = x\right)$.
- Backward induction (or dynamic programming):

$$X_1^* = \operatorname{argmin}_{x_1} \mathsf{E}_{0,x_1}\Bigl(\min_{x_2} \mathsf{E}_{1,x_2}\Bigl(\ldots \min_{x_N} \mathsf{E}_{N-1,x_N}\Bigl(\min_d \mathsf{E}_N\left(L(\xi, d)\right)\Bigr)\Bigr)\Bigr)$$

- Very difficult to use in practice beyond $N = 1$ or 2
  - each "min" is an optim. problem that needs to be solved...
  - each "$\mathsf{E}_{n,x}$" is an integral that needs to be computed...
  - none of them are tractable, even for the nicest (GP) priors ☹

Practical Bayesian optimization: myopic strategies

- Practical BO algorithms use, in general, myopic strategies
  - a.k.a. one-step look-ahead strategies
  - principle: make each decision as if it were the last one
  - Bayes-optimal if $N = 1$, sub-optimal otherwise

Practical Bayesian optimization: myopic strategies

- Practical BO algorithms use, in general, myopic strategies
  - a.k.a. one-step look-ahead strategies
  - principle: make each decision as if it were the last one
  - Bayes-optimal if $N = 1$, sub-optimal otherwise
- For any $n \leq N$, let $\overline{L}_n = \min_d \mathsf{E}_n\,(L(d))$

Practical Bayesian optimization: myopic strategies

- Practical BO algorithms use, in general, myopic strategies
  - a.k.a. one-step look-ahead strategies
  - principle: make each decision as if it were the last one
  - Bayes-optimal if $N = 1$, sub-optimal otherwise

- For any $n \leq N$, let $\overline{L}_n = \min_d \mathsf{E}_n\left(L(d)\right)$

Generic myopic BO algorithm

- For n from 0 to $N-1$
  - Compute $X_{n+1} = \operatorname{argmin}_x \mathsf{E}_{n,x_{n+1}}\left(\overline{L}_{n+1}\right)$
  - Make an evaluation at X_{n+1}
- Output $D_{N+1} = \operatorname{argmin} \mathsf{E}_N\left(L(d)\right)$

One-dimensional illustration (cont'd)

STK demo (https://github.com/stk-kriging/stk)

`stk_example_doe03`

- One-dimensional illustration
- Expected Improvement (EI) criterion
- Ordinary kriging, Matérn-5/2 covariance function (known parameters)

Practical Bayesian optimization: GP parameters

- Reminder: GP models have parameters
  - variance, range, etc.
  - "enough data" is needed to estimate them before the prior can usefully guide the sequential design
  - (alt.: introduce a prior distribution on the hyper-parameters)

Practical Bayesian optimization: GP parameters

- Reminder: GP models have parameters
  - variance, range, etc.
  - "enough data" is needed to estimate them before the prior can usefully guide the sequential design
  - (alt.: introduce a prior distribution on the hyper-parameters)

Generic myopic BO algorithm with hyper-parameter estimation

- Init: (space-filling) DoE of size $n0$ (rule of thumb: $n_0 = 10\,d$)
- For n from n_0 to $N-1$
  - once in a while, Estimate hyper-parameters (plug-in/fully Bayes)
  - Compute $X_{n+1} = \mathrm{argmin}_x\, \mathsf{E}_{n,x_{n+1}}\left(\overline{L}_{n+1}\right)$
  - Make an evaluation at X_{n+1}
- Output $D_{N+1} = \mathrm{argmin}\, \mathsf{E}_N\left(L(d)\right)$

Two-dimensional illustration

STK demo (https://github.com/stk-kriging/stk)

`demo1_EI`

- Two-dimensional illustration (Branin-Hoo)
- Expected Improvement (EI) criterion
- Ordinary kriging, Matérn-5/2 covariance function
- Parameters (variance, range) estimated by ReML

(This demo is not currently available in STK, the script will be provided directly to the participants as "supplementary material".)

Practical Bayesian optimization: optimization

- Each iteration involves an auxiliary optimization problem

Practical Bayesian optimization: optimization

- Each iteration involves an auxiliary optimization problem
- Various approaches to solve it
  - Fix grid or IID random search
    - OK for low-dimensional, simple problems
    - if accurate convergence is not needed

Practical Bayesian optimization: optimization

- Each iteration involves an auxiliary optimization problem
- Various approaches to solve it
  - Fix grid or IID random search
    - OK for low-dimensional, simple problems
    - if accurate convergence is not needed
  - External solvers
    - ex: DiceOptim $\rightarrow$ Rgenoud (genetic + gradient)
    - ex: Janusvekis & Le Riche (2013) $\rightarrow$ CMA-ES

Practical Bayesian optimization: optimization

- Each iteration involves an auxiliary optimization problem
- Various approaches to solve it
  - Fix grid or IID random search
    - OK for low-dimensional, simple problems
    - if accurate convergence is not needed
  - External solvers
    - ex: DiceOptim $\to$ Rgenoud (genetic + gradient)
    - ex: Janusvekis & Le Riche (2013) $\to$ CMA-ES
  - Sequential Monte Carlo (Benassi, 2013; Feliot et al, 2017)
    - sample according to a well-chosen sequence of densities

Practical Bayesian optimization: optimization

- Each iteration involves an auxiliary optimization problem
- Various approaches to solve it
  - Fix grid or IID random search
    - OK for low-dimensional, simple problems
    - if accurate convergence is not needed
  - External solvers
    - ex: DiceOptim $\rightarrow$ Rgenoud (genetic + gradient)
    - ex: Janusvekis & Le Riche (2013) $\rightarrow$ CMA-ES
  - Sequential Monte Carlo (Benassi, 2013; Feliot et al, 2017)
    - sample according to a well-chosen sequence of densities
- Bayesian optimization $\Rightarrow$ run-time overhead
  - depends on the model, sampling criterion, optimizer, etc.
  - BO is appropriate for expensive-to-evaluate numerical models