Bayesian optimization using sequential Monte Carlo

Romain Benassi, Julien Bect, and Emmanuel Vazquez

SUPELEC, Gif-sur-Yvette, France

Abstract. We consider the problem of optimizing a real-valued continuous function f using a Bayesian approach, where the evaluations of f are chosen sequentially by combining prior information about f, which is described by a random process model, and past evaluation results. The main difficulty with this approach is to be able to compute the posterior distributions of quantities of interest which are used to choose evaluation points. In this article, we decide to use a Sequential Monte Carlo (SMC) approach.

1 Overview of the contribution proposed

We consider the problem of finding the global maxima of a function $f : \mathbb{X} \to \mathbb{R}$, where $\mathbb{X} \subset \mathbb{R}^d$ is assumed bounded, using the *expected improvement* (EI) criterion [1, 3]. Many examples in the literature show that the EI algorithm is particularly interesting for dealing with the optimization of functions which are expensive to evaluate, as is often the case in design and analysis of computer experiments [2]. However, going from the general framework expressed in [1] to an actual computer implementation is a difficult issue.

The main idea of an EI-based algorithm is a Bayesian one: f is viewed as a sample path of a random process ξ defined on $\mathbb{R}^d$. For the sake of tractability, it is generally assumed that ξ has a Gaussian process distribution conditionally to a parameter $\theta \in \Theta \subseteq \mathbb{R}^s$, which tunes the mean and covariance functions of the process. Then, given a prior distribution π_0 on θ and some initial evaluation results $\xi(X_1), \ldots, \xi(X_{n_0})$ at $X_1, \ldots, X_{n_0}$, an (idealized) EI algorithm constructs a sequence of evaluations points $X_{n_0+1}, X_{n_0+2}, \ldots$ such that, for each $n \geq n_0$,

$$X_{n+1} = \operatorname*{argmax}_{x \in \mathbb{X}} \bar{\rho}_n := \int_{\theta \in \Theta} \rho_n(x; \theta) \mathrm{d}\pi_n(\theta)\,, \tag{1}$$

where π_n stands for the posterior distribution of θ, conditional on the σ-algebra $\mathcal{F}_n$ generated by $X_1, \xi(X_1), \ldots, X_n, \xi(X_n)$, and

$$\rho_n(x; \theta) := \mathsf{E}_{n,\theta}((\xi(X_{n+1}) - M_n)_+ \mid X_{n+1} = x)$$

is the EI at x given θ, with $M_n = \xi(X_0) \vee \cdots \vee \xi(X_n)$ and $\mathsf{E}_{n,\theta}$ the conditional expectation given $\mathcal{F}_n$ and θ. In practice, the computation of ρ_n is easily carried out (see [3]) but the answers to the following two questions will probably

have a direct impact on the performance and applicability of a particular implementation: a) How to deal with the integral in $\bar{\rho}_n$? b) How to deal with the maximization of $\bar{\rho}_n$ at each step?

We can safely say that most implementations—including the popular EGO algorithm [3]—deal with the first issue by using an *empirical Bayes* (or *plug-in*) approach, which consists in approximating π_n by a Dirac mass at the maximum likelihood estimate of θ. A plug-in approach using maximum a posteriori estimation has been used in [6]; *fully Bayesian* methods are more difficult to implement (see [4] and references therein). Regarding the optimization of $\bar{\rho}_n$ at each step, several strategies have been proposed (see, e.g., [3, 5, 7, 10]).

This article addresses both questions simultaneously, using a sequential Monte Carlo (SMC) approach [8, 9] and taking particular care to control the numerical complexity of the algorithm. The main ideas are the following. First, as in [5], a weighted sample $\mathfrak{T}_n = \{(\theta_{n,i}, w_{n,i}) \in \Theta \times \mathbb{R}, 1 \leq i \leq I\}$ from π_n is used to approximate $\bar{\rho}_n$; that is, $\sum_{i=1}^{I} w_{n,i}\ \rho_n(x; \theta_{n,i}) \to_I \bar{\rho}_n(x)$. Besides, at each step n, we attach to each $\theta_{n,i}$ a (small) population of candidate evaluation points $\{x_{n,i,j}, 1 \leq j \leq J\}$ which is expected to cover promising regions for that particular value of θ and such that $\max_{i,j} \bar{\rho}_n\,(x_{n,i,j}) \approx \max_x \bar{\rho}_n(x)$.

2 Algorithm and results

At each step $n \geq n_0$ of the algorithm, our objective is to construct a set of weighted particles

$$\mathfrak{G}_n = \Big\{\ \big(\gamma_{n,i,j}, w'_{n,i,j}\big)\ , \\ \gamma_{n,i,j} = (\theta_{n,i}, x_{n,i,j}) \in \Theta \times \mathbb{X}, w'_{n,i,j} \in \mathbb{R}\,,\ 1 \leq i \leq I,\, 1 \leq j \leq J\ \Big\} \quad (2)$$

so that $\sum_{i,j} w'_{n,i,j} \delta_{\gamma_{n,i,j}} \to_{I,J} \pi'_n$, with

$$\mathrm{d}\pi'_n(\gamma)\ =\ \tilde{g}_n(x \mid \theta)\,\mathrm{d}\lambda(x)\,\mathrm{d}\pi_n(\theta)\,, \quad x \in \mathbb{X}\,,\ \theta \in \Theta\,,\ \gamma = (\theta, x),$$

where λ denotes the Lebesgue measure, $\tilde{g}_n(x \mid \theta) = g_n(x \mid \theta)/c_n(\theta)$, $g_n(x \mid \theta)$ is a criterion that reflects the interest of evaluating at x (given θ and past evaluation results), and $c_n(\theta) = \int_{\mathbb{X}} g_n(x \mid \theta)\mathrm{d}x$ is a normalizing term. For instance, a relevant choice for g_n is to consider the probability that ξ exceeds M_n at x, at step n. (Note that we consider less θs than xs in $\mathfrak{G}_n$ to keep the numerical complexity of the algorithm low.)

To initialize the algorithm, generate a weighted sample $\mathfrak{T}_{n_0} = \{(\theta_{n_0,i}, w_{n_0,i}),$ $1 \leq i \leq I\}$ from the distribution π_{n_0}, using for instance importance sampling with π_0 as the instrumental distribution, and pick a density q_{n_0} over $\mathbb{X}$ (the uniform density, for example). Then, for each $n \geq n_0$:

Step 1: demarginalize — Using $\mathfrak{T}_n$ and q_n, construct a weighted sample $\mathfrak{G}_n$ of the form (2), with $x_{n,i,j} \overset{\text{iid}}{\sim} q_n$, $w'_{n,i,j} = w_{n,i} \frac{g_n(x_{n,i,j} \mid \theta_{n,i})}{q_n(x_{n,i,j}) c_{n,i}}$, and $c_{n,i} = \sum_{j'=1}^{J} \frac{g_n(x_{n,i,j'} \mid \theta_{n,i})}{q_n(x_{n,i,j'})}$.

Step 2: evaluate — Evaluate ξ at $X_{n+1} = \text{argmax}_{i,j} \sum_{i'=1}^{I} w_{n,i'} \rho_n(x_{n,i,j}; \theta_{n,i'})$.

Step 3: reweight/resample/move — Construct $\mathfrak{T}_{n+1}$ from $\mathfrak{T}_n$ as in [8]: reweight the $\theta_{n,i}$s using $w_{n+1,i} \propto \frac{\pi_{n+1}(\theta_{n,i})}{\pi_n(\theta_{n,i})} w_{n,i}$, resample (e.g., by multinomial resampling), and move the $\theta_{n,i}$s to get $\theta_{n+1,i}$s using an independant Metropolis-Hastings kernel.

Step 4: forge q_{n+1} — Form an estimate q_{n+1} of the second marginal of π'_n from the weighted sample $\mathfrak{X}_n = \{(x_{n,i,j}, w'_{n,i,j}), 1 \leq i \leq I, 1 \leq j \leq J\}$. Hopefully, such a choice of q_{n+1} will provide a good instrumental density for the next demarginalization step. Any (parametric or non-parametric) density estimator can be used, as long as it is easy to sample from; in this paper, a tree-based histogram estimator is used.

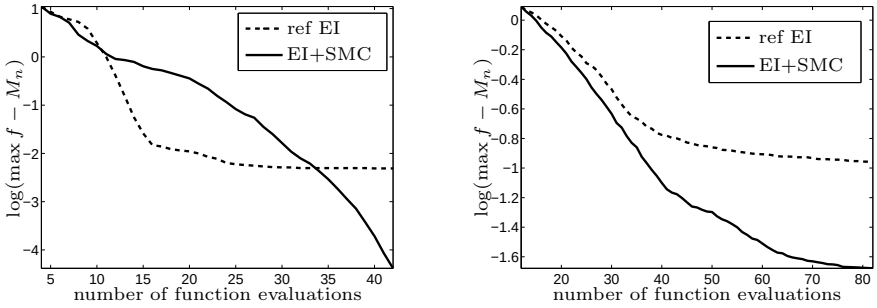
Nota bene: when possible, some components of θ are integrated out analytically in (1) instead of being sampled from; see [4].

Experiments. Preliminary numerical results, showing the relevance of a fully Bayesian approach with respect to empirical Bayes approach, have been provided in [4]. The scope of these results, however, was limited by a rather simplistic implementation (involving a quadrature approximation for $\bar{\rho}_n$ and a non-adaptive grid-based optimization for the choice of X_{n+1}). We present here some results that demonstrate the capability of our new SMC-based algorithm to overcome these limitations.

The experimental setup is as follows. We compare our SMC-based algorithm, with $I = J = 100$, to an EI algorithm in which: 1) we fix θ (at a "good" value obtained using maximum likelihood estimation on a large dataset); 2) X_{n+1} is obtained by exhaustive search on a fixed LHS of size $I \times J$. In both cases, we consider a Gaussian process ξ with a constant but unknown mean function (with a uniform distribution on $\mathbb{R}$) and an anisotropic Matérn covariance function with regularity parameter $\nu = 5/2$. Moreover, for the SMC approach, the variance parameter of the Matérn covariance function is integrated out using a Jeffreys prior and the range parameters are endowed with independent lognormal priors.

Results. Figures 1(a) and 1(b) show the average error over 100 runs of both algorithms, for the Branin function ($d = 2$) and the log-transformed Hartmann 6 function ($d = 6$). For the Branin function, the reference algorithm performs better on the first iterations, probably thanks to the "hand-tuned" parameters, but soon stalls due to its non-adaptive search strategy. Our SMC-based algorithm, however, quickly catches up and eventually overtakes the reference algorithm. On the Hartmann 6 function, we observe that the reference algorithm always lags behind our new algorithm.

We have been able to find results of this kind for other test functions. These findings are promising and need to be further investigated in a more systematic large-scale benchmark study.



(a) Branin function (dimension 2) (b) Hartmann 6 function (dimension 6)

Fig. 1. A comparison of the average error to the maximum (100 runs)

References

1. J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. In L. Dixon and G. Szego, editors, *Towards Global Optimization*, volume 2, pages 117–129. Elsevier, 1978.
2. T. J. Santner, B. J. Williams, W. I. Notz. The design and analysis of computer experiments, Springer, 2003.
3. D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *J. Global Optim.*, 13(4):455–492, 1998.
4. R. Benassi, J. Bect, and E. Vazquez. Robust Gaussian process-based global optimization using a fully Bayesian expected improvement criterion. In *LION5, online proceedings*, Roma, Italy, 2011.
5. R. Gramacy and N. Polson. Particle learning of Gaussian process models for sequential design and optimization, *J. Comput. Graph. Stat.*, 20(1):102–118, 2011.
6. D. J. Lizotte, R. Greiner and D. Schuurmans. An experimental methodology for response surface optimization methods, To appear in *J. Global Optim.*, 38 pages, 2011.
7. R. Bardenet and B. Kégl. Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm. In *ICML 2010, proceedings*, Haifa, Israel, 2010.
8. N. Chopin. A sequential particle filter method for static models, *Biometrika*, 89(3):539–552, 2002.
9. P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers, *J. R. Stat. Soc. B*, 68(3):411–436, 2006.
10. D. Ginsbourger and O. Roustant. DiceOptim: Kriging-based optimization for computer experiments, R package version 1.2, 2011.