A Bayesian Optimization Approach for Selecting the Best Parameters for Weighted Finite Difference Scheme Corresponding to Heat Equation

Zurab Kiguradze

Electromagnetic Compatibility Laboratory, Department of Electrical and Computer Engineering,
Missouri University of Science and Technology, Rolla, MO, USA
E-mail: kiguradzz@mst.edu

In the domain $[0;1]\times[0;T]$, where $T=const>0$, let us consider the initial-boundary value problem for the heat equation

$$\begin{gathered}\frac{\partial U(x,t)}{\partial t}-a\frac{\partial^2 U(x,t)}{\partial x^2}=f(x,t),\\ U(0,t)=U(1,t)=0,\ \ t\geq 0,\\ U(x,0)=U_0(x),\ \ x\in[0;1],\end{gathered}\tag{1}$$

where a is a positive constant and U_0 and f are given functions.

For the numerical solution of problem (1) let us introduce a net whose mesh points are denoted by $(x_i,t_j)=(ih,j\tau)$, where $i=0,1,\dots,M$; $j=0,1,\dots,N$ with $h=1/M$, $\tau=T/N$ and consider the following weighted finite difference scheme (see, for example, [8]):

$$\begin{gathered}\frac{u_i^{j+1}-u_i^j}{\tau}-a\Big[\sigma_1\,\frac{u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}}{h^2}+\sigma_2\,\frac{u_{i+1}^{j}-2u_i^{j}+u_{i-1}^{j}}{h^2}\Big]=\eta_1 f_i^{j+1}+\eta_2 f_i^j,\\ i=1,2,\dots,M-1;\ \ j=0,1,\dots,N-1,\\ u_0^j=u_M^j=0,\ \ j=0,1,\dots,N,\\ u_i^0=U_{0,i},\ \ i=0,1,\dots,M.\end{gathered}\tag{2}$$

Here the initial line is denoted by $j=0$. The discrete approximation at (x_i,t_j) is denoted by u_i^j and the exact solution to problem (1) at those points is denoted by U_i^j.

Qualitative and quantitative properties, as well as numerical solution for problem (1) and its nonlinear analogs are well studied in the literature (see, for example, [2, 3, 8] and the references therein). By tuning the parameters τ, h, σ_1, σ_2, η_1, η_2 and take relevant approximation for the right side the stability of the scheme (2), the different accuracy can be achieved for the numerical solution.

Our goal is to find the above-mentioned parameters automatically by using Bayesian machine learning. In particular, we will minimize objective function applying Bayesian Optimization (BO). The objective function is designed as a maximum of the absolute value of the difference between exact and numerical solutions at each grid point (x_i,t_j), $i=0,1,\dots,M$; $j=0,1,\dots,N$. For training, the different types of initial and boundary conditions with the corresponding right-hand side were selected. The output of the objective function depend on unknown parameters implicitly. Thus, we deal with, so-called black-box function optimization problem [1]. Since we do not have the close formula for the objective function, there is no information regarding gradient. So, the derivative-free optimization method is needed. BO is one of the most popular black-box optimization methods [1, 4–6]. It is based on Gaussian Process (GP) and Bayes Theorem [7]. BO is a model-based approach that makes sequential decisions to search the space, so the number of simulations gets minimized.

A GP is a generalization of the Gaussian Probability Distribution. Notation for Gaussian probability distribution is $\mathcal{N}(\mu,\sigma)$, where μ is mean and σ is standard deviation of random variables. While a Gaussian probability distribution describes random variables which are scalars or vectors, a stochastic process governs the properties of functions. GP is an extension of Multivariate Gaussian Distribution. In turn, the multivariate Gaussian distribution is a generalization of the one-dimensional normal distribution to higher dimensions. The probability density function of the multivariate Gaussian distribution in D-dimensions is defined by the following formula:

$$g(z)=\frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}}\exp\Big[-\frac{1}{2}\,(z-\mu)'\Sigma^{-1}(z-\mu)\Big],$$

where, in general Σ' denotes transpose of Σ, Σ^{-1} denotes inverse of Σ, $\mu=(\mu_1,\mu_2,\dots,\mu_D)$ is mean vector of $z=(z_1,z_2,\dots,z_D)$ and $\Sigma=\operatorname{cov}[z]$ is the $D\times D$ covariance matrix, which is positively defined and is constructed by, one of the so-called covariance functions [7]. One of the common covariance function is Squared Exponential function:

$$K_{SE}=k(z_i,z_j)=\sigma\exp\Big(-\frac{1}{2}\,\frac{\|z_i-z_j\|^2}{\theta^2}\Big),$$

where σ and θ are hyper-parameters which can be tuned by users. Note that GP is fully described by mean and covariance functions.

Most of the efficiency derived from Bayesian optimization ability to incorporate prior belief about the problem to help direct the sampling, and to trade of exploration and exploitation of the search space [1]. Algorithm is called Bayesian because it uses the well-known Bayes Theorem, which can be stated as follows

$$P(A\mid B)\sim P(B\mid A)P(A),$$

where $P(A\mid B)$ is probability of A given B, $P(B\mid A)$ is probability of B given A and $P(A)$ is the marginal probability [1,7].

Let us now consider how the new query can be obtained using the aforementioned Bayes Theorem. Assume, the dataset with n points is already obtained $D_{1:n}=\{z_{1:n},g(z_{1:n})\}$. Bayes Theorem helps to estimate posterior distribution $P(g\mid D_{1:n})$ by combining a prior distribution $P(g)$ with the likelihood function $P(D_{1:n}\mid g)$

$$P(g\mid D_{1:n})\sim P(D_{1:n}\mid g)P(g).$$

To find the next sample point z_{n+1}, the so-called acquisition function is maximized. There are different types of acquisition functions. One of the most popular acquisition function is Upper Confidence Bound (UCB)

$$\mathrm{UCB}\,(z)=\mu(z)+\kappa\sigma(z),$$

where κ is tunable trade-off parameter.

The BO algorithm performs as follows:

1. Collect data $D_{1:n}=\{z_{1:n},g(z_{1:n})\}$ and fit the GP. Note, that BO can be started from one point dataset;

2. Find the next querying point by maximizing acquisition function;

3. Augment dataset $D_{1:n+1}=\{z_{1:n+1},g(z_{1:n+1})\}$ and update GP;

4. End process when the desired accuracy is obtained or the number of iterations reaches a certain value.

Note that all steps in the BO algorithm are clear except step 3 (note also that since the evaluation of the acquisition function is not expensive its maximization in step 2 can be done by some standard optimization algorithm). In step 3 we need to update the GP and find the updated mean and variance functions, based on which the acquisition functions are constructed. Bellow, the close formulas for calculating the updated mean and variance functions are given. Assume, dataset $D_{1:n} = \{z_{1:n}, g(z_{1:n})\}$ is already obtained. The function values are drawn according to a multivariate normal distribution $\mathcal{N}(0, K)$, where the kernel matrix is given by:

$$K = \begin{bmatrix} k(z_1, z_1) & \cdots & k(z_1, z_n) \\ \vdots & \ddots & \vdots \\ k(z_n, z_1) & \cdots & k(z_n, z_n) \end{bmatrix}.$$

Let us denote $g_{n+1} = g(z_{n+1})$, where z_{n+1} is the next sampling point, which is obtained from the maximization of the acquisition function. g_{n+1} and $g_{1:n}$ are jointly Gaussian:

$$\begin{bmatrix} g_{1:n} \\ g_{n+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K & k \\ k' & k(z_{n+1}, z_{n+1}) \end{bmatrix}\right),$$

where

$$k = \big[k(z_{n+1}, z_1), k(z_{n+1}, z_2), \dots, k(z_{n+1}, z_n)\big].$$

Using the Sherman–Morrison–Woodbury formula [1,7] the following predictive distribution can be obtained:

$$P(g_{n+1} \mid D_{1:n+1}, z_{n+1}) \sim \mathcal{N}\big(\mu_n(z_{n+1}), \sigma_n^2(z_{n+1})\big),$$

where

$$\begin{aligned} \mu_n(z_{n+1}) &= k'K^{-1}g_{1:n}, \\ \sigma_n^2(z_{n+1}) &= k(z_{n+1}, z_{n+1}) - k'K^{-1}k. \end{aligned}$$

To implement the BO for our problem the IMGPO (Infinite-Metric GP Optimization) algorithm is used [4]. Note that the IMGPO algorithm does not require any prior data. It can be started from any random point, say from the center point of the search space, as in our case. IMGPO uses *UCB* acquisition function and avoids its maximization for finding the next sample point, instead it handles the tradeoff with the assumption of the existence of a tighter bound than *UCB* and remain the exponential convergence at the same time (for details see [4]).

In our numerical experiment we took $\sigma_2 = 1 - \sigma_1$, $\eta_2 = 1 - \eta_1$ and the search space is $(\tau, h, \sigma_1, \eta_1) \in [0; 0.1] \times [0; 0.1] \times [0; 1] \times [0; 1]$. The stopping criterion for BO is as follows, the algorithm stops when maximum error is less then $\tau + h^2$ and less than $\varepsilon = 0.0001$ or maximum error is less than $\tau + h^2$ and maximum number of iterations $I = 30$ is reached.

We have carried out various numerical experiments for different test cases and found the values of parameters for the best performance of the scheme (2).

References

[1] E. Brochu, V. M. Cora and N. de Freitas, A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *ArXiv preprint*, arXiv:1012.2599, 2010; `https://arxiv.org/abs/1012.2599`.

[2] T. Jangveladze, Investigation and numerical solution of nonlinear partial differential and integro-differential models based on system of Maxwell equations. *Mem. Differ. Equ. Math. Phys.* **76** (2019), 1–118.

[3] T. Jangveladze, Z. Kiguradze and B. Neta, *Numerical Solutions of Three Classes of Nonlinear Parabolic Integro-Differential Equations.* Elsevier/Academic Press, Amsterdam, 2016.

[4] K. Kawaguchi, L. P. Kaelbling and T. Lozano-Pérez, Bayesian optimization with exponential convergence. *Proceeding NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems* **2** (2015), 2809–2817.

[5] Z. Kiguradze, J. He, B. Mutnury, A. Chada and J. Drewniak, Bayesian optimization for stack-up design. *2019 IEEE International Symposium on Electromagnetic Compatibility, Signal & Power Integrity (July 22–26, 2019)*, pp. 629–634, New Orleans, LA, 2019.

[6] J. Mockus, *Bayesian Approach to Global Optimization. Theory and Applications. With a 5.25-inch IBM PC Floppy Disk.* Mathematics and its Applications (Soviet Series), 37. Kluwer Academic Publishers Group, Dordrecht, 1989.

[7] C. E. Rasmussen and Ch. K. I. Williams, *Gaussian Processes for Machine Learning.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006.

[8] A. A. Samarskiĭ, *Theory of Difference Schemes.* (Russian) Nauka, Moscow, 1977.