Bayesian Global Optimization

Contents

References

- A Tutorial on Bayesian Global Optimization, by Peter Frazier

Baysian Global Optimization of a Scalar Function without Noise

We are going to addresss the problem:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} f(\mathbf{x}).$$

under the assumption that:

- we can evaluate $f(\mathbf{x})$ at any $\mathbf{x}$;
- evaluating $f(\mathbf{x})$ takes a lot of time/money;
- we cannot evaluate the gradient $\nabla f(\mathbf{x})$;
- the dimensionality of $\mathbf{x}$ is not very high.

Why is this an important problem? Well, $\mathbf{f}(\mathbf{x})$ may be a very expensive simulation that you would like to optimize under a limitted budget. It could also be an experimentally measured quantity of interest, but you would have to make sure that the noise is not very big. We will see what you can do when you have noise later.

Sequential Information Acquisition for Optimization

What do we mean by *sequential information acquisiton* for optimization? It is just an algorithm that tells us where to evaluate a function next if we would like to find its maximum/minimum. The problem can be formulated in a rigorous mathematical way using the theory and tools of dynamic programming. However, the full fledged mathematical problem of designing optimal evaluation policies tends to be harder than the original problem of just maximizing a function. So, we usually resort to heurestics. These heuristics are not optimal in a mathematical sense, but they are convenient in a practical sense. In a way, the heuristics strike a balance the theoretically optimal (but computationally intractable) thing to do, and just randomly selecting evaluations.

A generic heuristic, which we are going to call "one-step-look ahead information acquisition policy" works as follows:

- We start with an initial data set consiting of n_0 input-output observations:

$$\mathcal{D}_{n_0} = (\mathbf{x}_{1:n_0}, \mathbf{y}_{1:n_0}).$$

For example, one may obtain these through Latin Hypercube Sampling of their input space.

- For $n = n_0, n_0 + 1, \ldots$, we start do the following:
  - We use the current dataset to quantify our state of knowledge about $f(\mathbf{x})$. For example, we can use Gaussian process regression (GPR) or any other Bayesian regression method to obtain the predictive distribution:
$$f(\cdot)|\mathcal{D}_{n_0} \sim p(f(\cdot)|\mathcal{D}_{n_0}).$$
  - Then we pick the *most important* point to evaluate next by looking at maximizing an *acquisition function* $a_{n_0}(\mathbf{x})$ which depends on our current state of knowledge (we will see specific examples below). This acquisition function quantifies, for example, how much value or how much information there is in in doing an evaluation at $\mathbf{x}$. We can assume that it is a nonnegative function. So, we to

pick the next point we solve a problem of the form:

$$\mathbf{x}_{n+1} = \arg\max a_n(\mathbf{x}).$$

That is, we pick the point with the maximum value or the maximum information.

- If maximum value of the acquisition function is smaller than a threshold, e.g., if $a_n(\mathbf{x}) \leq \epsilon$ for some $\epsilon > 0$, then we STOP.
- Otheweise, we evaluate the function at the selected $\mathbf{x}_{n+1}$ to obtain:

$$y_n = f(\mathbf{x}_n).$$

i.e., if the maximum value is below some threshold (not enough value added to continue search) - close enough to the max being searched for

This would entain either running a simulation or doing an experiment.

- We augment our original data set with the new observation:

$$\mathcal{D}_{n+1} = ((\mathbf{x}_{1:n}, \mathbf{x}_n), (\mathbf{y}_{1:n}, y_n)).$$

- We use Bayes' rule to update our state of knowledge:

$$f(\cdot)|\mathcal{D}_{n+1} \sim p(f(\cdot)|\mathcal{D}_{n+1}) \propto p(y_{n+1}|x_{n+1}, f(\cdot))p(f(\cdot)|\mathcal{D}_n).$$

- If we have run out of evaluation budget, we STOP. Otherwise, we continue the loop.

- At this point, we report our current state of knowledge about the maximum of the function. For example, we can report what is the *observed maximum*. That is, we can find the index of the evaluation corresponding to the observed maximum:

$$i^* = \arg\max_{1 \leq i \leq n} y_i,$$

and say that the maximum was y_{i^*} and the location of the maximum is $\mathbf{x}_{i^*}$. Alternatively, and in particular if our optimizatioin budget is low, we can also quantify our epistemic uncertainty about the location of the maximum. We discuss this idea in detail towards the end of this notebook.

We will provide specific examples of information acquisition functions in the hands-on activities.

By Ilias Bilionis (ibilion[at]purdue.edu)
© Copyright 2021.