Bayesian Optimization

September 25, 2021

1 First Order Bayesian Optimization

First Order Bayesian Optimization (FOBO) deals with utilizing the gradient information along with the function value to find the maximum value of the function. One possible way to utilize the gradient information is to take advantage of the fact that the gradient vanishes at the maxima i.e. $\nabla f(x) = 0$ when $f(x)$ is maximum. In other words, we search for points x such that $\nabla f(x) = 0$.

1.1 Independent Surrogate Gaussian Processes

If we relax the joint assumption between the objective function and its partial derivatives and model each of them using an independent Gaussian Process, then we would have (d+1) GPs, where d represents the dimension of the objective function. Mathematically,

$$f(\cdot) \sim GP(\mu(\cdot), K(\cdot, \cdot)) \tag{1}$$

$$\frac{\partial f(\cdot)}{\partial x(i)} \sim GP(\mu_i(\cdot), K_i(\cdot, \cdot)) \tag{2}$$

Therefore, we model the objective function and its partial derivatives using separate mean and kernel functions. This way we can parallelize the fitting of GPs.

1.2 Acquisition Algorithms

Since, at the point of maxima the gradient vanishes, the objective function should next be queried at these points. Therefore, we should try to minimize the expected value of absolute partial derivative. We can define a new utility function for each of the partial derivative GP as follows:

$$I_i(\mathbf{x}) = \mathbb{E}_n\left(\left|\frac{\partial f(\cdot)}{\partial x(i)}\right|\right) i \in 1, ..., d. \tag{3}$$

Therefore, the next query point is given by,

$$\mathbf{x}_i^{n+1} = arg \min_{\mathbf{x} \in D} I_i(\mathbf{x}), i \in 1, ..., d. \tag{4}$$

Now, we will have d+1 suggestions for the next query point (d from each of the partial derivative GP model and 1 from the function GP model). The next query point can be obtained using any of the following alternatives:

- In the first way, the information is aggregated by taking a weighted convex combination of all the points suggested i.e.,

$$\mathbf{x}^{n+1} = \sum_{i=0}^{d} \frac{exp(\mu^{(n)}(\mathbf{x}_i^{n+1}))}{\sum_{i=0}^{d} exp(\mu^{(n)}(\mathbf{x}_i^{n+1}))} \mathbf{x}_i^{n+1}. \tag{5}$$

- In the second way, the information is aggregated by taking the point that has the maximum significance i.e.,

$$\begin{aligned} i^* &= arg\max_i \mu^{(n)}(\mathbf{x}_i^{n+1}), \\ \mathbf{x}^{n+1} &= \mathbf{x}_{i^*}^{n+1} \end{aligned} \tag{6}$$

2 Experiments

1. **Best of both the worlds**
   Instead of using either (5) or (6), we can combine both and pick the point which has the highest mean. Let the point suggested by (5) and (6) be $\mathbf{x}_{con}^{n+1}$ and $\mathbf{x}_{max}^{n+1}$. Then, the next query point suggested would be,

$$\mathbf{x}^{n+1} = argmax\{\mu^{(n)}(\mathbf{x}_{con}^{n+1}), \mu^{(n)}(\mathbf{x}_{max}^{n+1})\} \tag{7}$$

   where the mean is calculated using the function GP model.

2. **cUpper**
   We slightly modify (5) to also include the variance of the objective function value at the suggested points. This will help to make the search little bit explorative in nature rather than being purely exploitative in nature. The idea is similar to Upper Confidence Bound acquisition function. Mathematically, the new equation would be,

$$\mathbf{x}^{n+1} = \sum_{i=0}^{d} \frac{exp(\mu^{(n)}(\mathbf{x}_i^{n+1}) + \alpha\sigma^{(n)}(\mathbf{x}_i^{n+1}))}{\sum_{i=0}^{d} exp(\mu^{(n)}(\mathbf{x}_i^{n+1}) + \alpha\sigma^{(n)}(\mathbf{x}_i^{n+1}))} \mathbf{x}_i^{n+1}. \tag{8}$$

   The hyperparameter α can be decreased gradually using a temperature schedule to make the search more exploitative. In the experiment $\alpha = 1$ was used.

3. **(3) + variance**
   We slightly modify (3) to also include the variance of the absolute partial derivative. This way we can eliminate those points which although have mean absolute partial derivative close to 0 but also have high variance.

In other words, this would make the search more exploitative in nature. Mathematically,

$$I_i(\mathbf{x}) = \mathbb{E}_n\left(\left|\frac{\partial f(\cdot)}{\partial x(i)}\right|\right) + \beta\sigma_n\left(\left|\frac{\partial f(\cdot)}{\partial x(i)}\right|\right) i \in 1, ..., d. \tag{9}$$

Therefore, the next query point is given by,

$$\mathbf{x}_i^{n+1} = arg\min_{\mathbf{x}\in D} I_i(\mathbf{x}), i \in 1, ..., d. \tag{10}$$

The hyperparameter β can be tuned using a temperature schedule. In the experiment, $\beta = 1$ was used.
Once, we obtain $(d+1)$ suggestions, we combine them using (5), to get the next query point.

4. In this experiment we ignore the point suggested by the function GP model while taking the convex combination in (5). In other words, while taking the convex combination we just consider the points suggested by the d partial derivative GP models. This way we try to find how important or useful is the gradient information. Mathematically,

$$\mathbf{x}^{n+1} = \sum_{i=1}^{d} \frac{exp(\mu_{(n)}(\mathbf{x}_i^{n+1}))}{\sum_{i=1}^{d} exp(\mu_{(n)}(\mathbf{x}_i^{n+1}))}\mathbf{x}_i^{n+1}. \tag{11}$$

5. We suggest another method apart from (5) and (6), to combine the points suggested by $(d+1)$ GP models. In this method, we try to just leverage the gradient information and hence ignore the point suggested by the function GP model. We calculate the mean and variance of partial derivatives at each of the remaining d suggested point using the d partial derivative GPs. For each of the d we sum the mean and variance of partial derivative and stack them into a vector. This vector can be thought of as the gradient of the objective function at the location. So, we choose the point with the lowest l2-norm of this gradient vector as our next query point.

$$\frac{\partial f(\cdot)}{\partial \mathbf{x}_i^{n+1}} = \begin{bmatrix} \frac{\partial f(\cdot)}{\partial x_i^{n+1}(1)} \\ \frac{\partial f(\cdot)}{\partial x_i^{n+1}(2)} \\ \vdots \\ \frac{\partial f(\cdot)}{\partial x_i^{n+1}(d)} \end{bmatrix} \approx \begin{bmatrix} \mu_1^{(n)}(\mathbf{x}_i^{n+1}) + \sigma_1^{(n)}(\mathbf{x}_i^{n+1}) \\ \mu_2^{(n)}(\mathbf{x}_i^{n+1}) + \sigma_2^{(n)}(\mathbf{x}_i^{n+1}) \\ \vdots \\ \mu_d^{(n)}(\mathbf{x}_i^{n+1}) + \sigma_d^{(n)}(\mathbf{x}_i^{n+1}) \end{bmatrix} \tag{12}$$

where $\mu_j^{(n)}(\cdot)$ and $\sigma_j^{(n)}(\cdot)$ represents the mean and variance function of the jth dimension partial derivative. Therefore, the next query point is given by,

$$\mathbf{x}^{n+1} = arg\min_{i}\left|\left|\frac{\partial f(\cdot)}{\partial \mathbf{x}_i^{n+1}}\right|\right|_2 \tag{13}$$

6. Instead of having $(d+1)$ GPs for the objective function and its partial derivatives, we have a single GP for

$$-f(x) + \gamma \left|\left| \frac{\partial f(\cdot)}{\partial \mathbf{x}} \right|\right|_1 \tag{14}$$

This way we can increase the steepness of the original objective function and therefore, reach to the optimal value faster. However, since the addition of norm of the gradient can create artificial maximas, we use a temperature schedule for γ so that eventually the influence of the norm of the gradient term becomes zero. The temperature schedule used in the experiment was

$$\gamma_t = \gamma_0 * \delta^t \tag{15}$$

where t represents the number of iterations, γ_t represents the value of γ after t iterations. The value of γ_0 and δ was set to 1 and 0.95 respectively.

7. Instead of using mean in (5) and (6), we use the Expected Improvement of the point $\mathbf{x}_i^{n+1}$ to calculate the convex combination or obtianing the point with maximum significance. Mathematically, (5) transforms into,

$$\mathbf{x}^{n+1} = \sum_{i=0}^{d} \frac{exp(EI^{(n)}(\mathbf{x}_i^{n+1}))}{\sum_{i=0}^{d} exp(EI^{(n)}(\mathbf{x}_i^{n+1}))} \mathbf{x}_i^{n+1}. \tag{16}$$

and (6) transforms into,

$$\begin{aligned} i^* = arg \max_i EI^{(n)}(\mathbf{x}_i^{n+1}), \\ \mathbf{x}^{n+1} = \mathbf{x}_{i^*}^{n+1} \end{aligned} \tag{17}$$

Note, that the Expected Improvement is calculated using the objective function GP model.