Recent Works on Batch Bayesian Optimization

Jungtaek Kim (jtkim@postech.ac.kr)

Machine Learning Group,
Department of Computer Science and Engineering, POSTECH,
77 Cheongam-ro, Nam-gu, Pohang 37673,
Gyeongsangbuk-do, Republic of Korea

October 9, 2018

Table of Contents

Batch Bayesian Optimization

Motivation

- Bayesian optimization usually solves expensive-to-evaluate function
- Evaluation a true function is one of bottlenecks of Bayesian optimization
- Batch Bayesian optimization (BBO) is a Bayesian optimization method for querying n_b points simultaneously
- Using BBO, acquired points can be evaluated in parallel

Selected Methods of BBO

- Random: Acquiring one point chosen by acquisition function and $n_b - 1$ randomly chosen points
- Constant: Acquiring n_b points using surrogate function with constant fake observations
- Prediction: Acquiring n_b points using surrogate function with observations which produce outputs as posterior mean function over the corresponding covariates
- Partially update
- Hybrid approach
- Local penalizer

Global versus Local Search in Constrained Optimization of Computer Models

- Matthias Schonlau, William J. Welch, and Donald R. Jones (1998)
- Update variance term without obtaining responses:

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, \mathbf{X}) \left[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} \right]^{-1} K(\mathbf{X}, \mathbf{x}^*)$$

Hybrid Batch Bayesian Optimization

- Javad Azimi, Ali Jalali, Xiaoli Z. Fern (2012)
- ICML-2012
- A hybrid algorithm that dynamically switches between sequential and batch with variable batch sizes
- The authors provide theoretical justification of their algorithm

Hybrid Batch Bayesian Optimization

Algorithm 1 Hybrid Batch Expected Improvement

Input: Total budget of experiments (n_l), maximum batch size (n_b), the predictor ($\widehat{y}$), current observation $\mathcal{O} = (\boldsymbol{x}_{\mathcal{O}}, \boldsymbol{y}_{\mathcal{O}})$ and stopping threshold ϵ.

while $n_l > 0$ **do**
- $x_1^* \leftarrow \arg\max_{x \in \mathcal{X}} \; EI(x|\mathcal{O})$.
- $\mathcal{A} \leftarrow (x_1^*, \hat{y}_1), \quad n_l \leftarrow n_l - 1$.
- $z \leftarrow \arg\max_{x \in \mathcal{X}} \; \widehat{EI}(x|\mathcal{O} \cup \mathcal{A})$.
- **while** $\left(\gamma_z(\theta_{\boldsymbol{x}_{\mathcal{A}}} + \|\hat{y}_{\mathcal{A}} - \mu_{\boldsymbol{x}_{\mathcal{A}}|\mathcal{O}}\|_2) \le \epsilon\right)$ and $(n_l > 0)$ and $(|\mathcal{A}| < n_b)$ **do**
  - $\mathcal{A} \leftarrow \mathcal{A} \cup (z, \hat{y}_z), \quad n_l \leftarrow n_l - 1$.
  - $z \leftarrow \arg\max_{x \in \mathcal{X}} \; \widehat{EI}(x|\mathcal{O} \cup \mathcal{A})$.
- **end while**
- $\boldsymbol{y}_{\mathcal{A}} \leftarrow \text{RunExperiment}(\boldsymbol{x}_{\mathcal{A}})$
- $\mathcal{O} \leftarrow \mathcal{O} \cup (\boldsymbol{x}_{\mathcal{A}}, \boldsymbol{y}_{\mathcal{A}})$

end while

return $\max(\boldsymbol{y}_{\mathcal{O}})$

Hybrid Batch Bayesian Optimization

Definition 1. *Let* $\boldsymbol{x} = \{x_1, x_2, \ldots, x_m\} \in \mathcal{X} \setminus \boldsymbol{x}_{\mathcal{O}}$ *be any unobserved set of points. Let* $\widehat{\boldsymbol{y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m\}$ *be our estimate of their outputs based on GP considering* $y_i|\mathcal{O} \sim \mathcal{N}(\mu_{x_i|\mathcal{O}}, \sigma^2_{x_i|\mathcal{O}})$*. For any new point* $z \in \mathcal{X} \setminus \{\boldsymbol{x}_{\mathcal{O}} \cup \boldsymbol{x}\}$*, let* $y_z|\mathcal{O} \sim \mathcal{N}(\mu_{z|\mathcal{O}}, \sigma^2_{z|\mathcal{O}})$ *and* $y_z|\mathcal{O}, (\boldsymbol{x}, \widehat{\boldsymbol{y}}) \sim \mathcal{N}(\widehat{\mu}_{z|\mathcal{O},\boldsymbol{x}}, \widehat{\sigma}^2_{z|\mathcal{O},\boldsymbol{x}})$*.*

Hybrid Batch Bayesian Optimization

Theorem 1. *Assuming* $\Delta(\sigma_z) := \sigma^2_{z|\mathcal{O}} - \sigma^2_{z|\mathcal{O},\boldsymbol{x}}$, *we have*

$$\Delta(\sigma_z) = \left(CA^{-1}B^T - k(z,\boldsymbol{x})\right) D \left(CA^{-1}B^T - k(z,\boldsymbol{x})\right)^T, \tag{2}$$

where, $B = k(\boldsymbol{x}, \boldsymbol{x}_{\mathcal{O}})$, $A = k(\boldsymbol{x}_{\mathcal{O}}, \boldsymbol{x}_{\mathcal{O}})$, $C = k(z, \boldsymbol{x}_{\mathcal{O}})$ *and* $D = (k(\boldsymbol{x}, \boldsymbol{x}) - BA^{-1}B^T)^{-1}$.

Hybrid Batch Bayesian Optimization

Theorem 2. *Let* $\gamma_z = \left\|(k(z, \boldsymbol{x}) - CA^{-1}B^T)D\right\|_2$. *Then,*

$$\left|\mu_{z|\mathcal{O},\boldsymbol{x}} - \widehat{\mu}_{z|\mathcal{O},\boldsymbol{x}}\right| \leq \gamma_z \left\|\boldsymbol{y} - \widehat{\boldsymbol{y}}\right\|_2$$

$$\left|\mu_{z|\mathcal{O},\boldsymbol{x}} - \mu_{z|\mathcal{O}}\right| \leq \gamma_z \left\|\boldsymbol{y} - \mu_{\boldsymbol{x}|\mathcal{O}}\right\|_2.$$

Hybrid Batch Bayesian Optimization

Corollary 1. *Let* $\theta_{\boldsymbol{x}} := \sqrt{\sum_{i=1}^{m} \sigma^2_{x_i|\mathcal{O}}}$, *then*

$$\mathbb{E}_{\boldsymbol{y}}\left[\left|\mu_{z|\mathcal{O},\boldsymbol{x}} - \mu_{z|\mathcal{O}}\right|\right] \leq \gamma_z \theta_{\boldsymbol{x}}.$$

Moreover,

$$\mathbb{E}_{\boldsymbol{y}}\left[\left|\mu_{z|\mathcal{O},\boldsymbol{x}} - \widehat{\mu}_{z|\mathcal{O},\boldsymbol{x}}\right|\right] \leq \gamma_z \left(\theta_{\boldsymbol{x}} + \|\widehat{y} - \mu_{\boldsymbol{x}|\mathcal{O}}\|_2\right).$$

Batch Bayesian Optimization via Local Penalization

- Javier Gonzalez, Zhenwen Dai, Philipp Hennig, and Neil Lawrence (2016)
- AISTATS-2016
- A method to penalize local region using L-Lipschitz cone
- The authors provide a method to select L and M

Batch Bayesian Optimization via Local Penalization

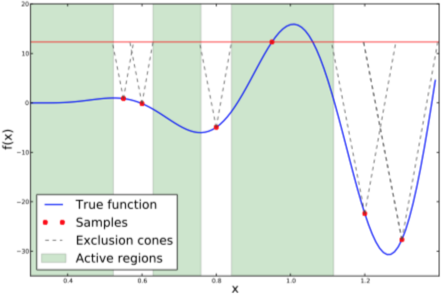

Figure 1: Forrester function $f(x) = (6x-2)^2 \sin(12x-4)$ in the interval $[0.3, 01, 4]$. We take 6 evaluations $\mathbf{x}_1, \ldots, \mathbf{x}_6$ of the function, $M = \max_i f(\mathbf{x}_i)$ and $L = 400$. The exclusion zones for the maximum of f determined by the balls $B_r(\mathbf{x}_i)$ are shown.

Batch Bayesian Optimization via Local Penalization

Proposition 1 *Let $f(\mathbf{x})$ be a $\mathcal{GP}$ with posterior mean $\mu_n(\mathbf{x})$ and posterior variance $\sigma_n^2(\mathbf{x})$. The function $\varphi(\mathbf{x}; \mathbf{x}_j)$ in Eq. (6) is a valid local penalizer of $\alpha(\mathbf{x})$ at $\mathbf{x}_j$ such that:*

$$\varphi(\mathbf{x}; \mathbf{x}_j) = \frac{1}{2} erfc\,(-z)$$

where $z = \frac{1}{\sqrt{2\sigma_n^2(\mathbf{x}_j)}} \left(L\|\mathbf{x}_j - \mathbf{x}\| - M + \mu_n(\mathbf{x}_j) \right)$, *for erfc the complementary error function,* $M = \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ *and L a valid Lipschitz constant.*

Batch Bayesian Optimization via Local Penalization

- $$\hat{M} = \max_{\mathcal{X}} \mu_n \mathbf{x} \quad \text{or} \quad \hat{M} = \max_{i} \{y_i\}$$

- $$\begin{aligned} \hat{L} &= \max_{\mathcal{X}} \|\mu_\nabla(\mathbf{x}^*)\| \\ &= \max_{\mathcal{X}} \|\partial K(\mathbf{x}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n \mathbf{I}]^{-1} \mathbf{y}\| \end{aligned}$$

Batch Bayesian Optimization via Local Penalization

Algorithm 1 Batch Bayesian Optimization with Local Penalization (BBO-LP).

Input: dataset $\mathcal{D}_1 = \{\mathbf{x}_i, y_i\}_{i=1}^n$, batch size n_b, iteration budget m, acquisition transformation g.
for $t = 1$ **to** m **do**
  Fit a GP to $\mathcal{D}_t$ and the acquisition function $\alpha(\mathbf{x}, \mathcal{I}_{t,0})$.
  $\tilde{\alpha}_{t,0}(\mathbf{x}) \leftarrow g(\alpha(\mathbf{x}, \mathcal{I}_{t,0}))$.
  $\hat{L} \leftarrow \max_{\mathcal{X}} \|\mu_{\nabla}(\mathbf{x})\|$.
  for $j = 1$ **to** n_b **do**
    1. *M-step:* $\mathbf{x}_{t,j} \leftarrow \arg\max_{x \in \mathcal{X}} \{\tilde{\alpha}_{t,j-1}(\mathbf{x})\}$.
    2. *P-step:* $\tilde{\alpha}_{t,j}(\mathbf{x}) \leftarrow \tilde{\alpha}_{t,0}(\mathbf{x}) \prod_{j=1}^{k} \varphi(\mathbf{x}; \mathbf{x}_{t,j}, \hat{L})$.
  end for
  $\mathcal{B}_t^{n_b} \leftarrow \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,n_b}\}$.
  $y_{t,1}, \ldots, y_{t,n_b} \leftarrow$ Parallel evaluations of f at $\mathcal{B}_t^{n_b}$.
  $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\mathbf{x}_{t,j}, y_{t,j})\}_{j=1}^{n_b}$.
end for
Fit GP to $\mathcal{D}_n$.
Returns: $\hat{\mathbf{x}}_M = \arg\max_{x \in \mathcal{X}} \{\mu(\mathbf{x})\}$.