Bayesian Optimization with adaptive discretization

Black-box optimization

Consider the problem of finding a maximizer of a function $f : \mathcal{X} \to \mathbb{R}$.

Assumptions:

- f is not known explicitly; can only be accessed through evaluation queries.
- The observations of f are noisy.
- The function f is expensive to evaluate.

Goal: Design a sequential strategy of selecting query points to quickly reach a global optimizer of f.

Regularity assumptions on f

- We want to get close to the global optimizer x^* from finite number (n) of observations.
- Informally we require that:
  - Evaluating f at some $x \in \mathcal{X}$ gives some information about f in its neighbourhood.
  - Finitely many such neighborhoods cover whole of $\mathcal{X}$.
- Two common ways of imposing these conditions:
  1. Lipschitz Optimization: Explicit smoothness assumptions on f
  2. Bayesian Framework: f is a sample from a stochastic process

Problem Setting

- Suppose $f : \mathcal{X} \to \mathbb{R}$ is a sample from $GP(0, K)$.
- Observation model: $y = f(x) + \eta$ with $\eta \sim N(0, \sigma^2)$.
- $\mathcal{X}$ is a compact subset of $\mathbb{R}^D$
- Budget $= n$ evaluations
- Select queries $\{X_1, X_2, \ldots, X_n\}$ sequentially
- Performance measures:
  - Simple regret: $\mathcal{S}_n = f(x^*) - f(x_n^*)$
  - Cumulative regret: $\mathcal{R}_n = \sum_{t=1}^{n} f(x^*) - f(x_t)$

Example-1: Hyperparameter tuning in ML models

- Suppose the learning algorithm with hyperparameters θ outputs classifier $A(\theta)$
  - $\mathcal{X}=$ space of hyperparameters.
  - $f(\theta)=$ performance of $A(\theta)$ on test dataset.
  - Constraint: finite computational resources $\Rightarrow n$-attempts.
- Goal:After n rounds, Output θ_n^*: our best guess
- Performance metric: **Simple Regret**

$$\mathcal{S}_n = f(\theta^*) - f(\theta_n^*)$$

- Pure *exploration* problem.

Example-2: Clinical Trials

- $\mathcal{X} = \{t_1, t_2, \ldots, t_m\}$ = set of all possible treatments.
- f = response of patients to a particular treatment.
- n = number of patients available for trial.
- Goal: Find the best strategy of assigning treatments while minimizing harm to patients.
- Performance metric: **Cumulative Regret**

$$\mathcal{R}_n = \sum_{i=1}^{n} f(t^*) - f(t_i)$$

- Presents an *exploration-exploitation* dilemma.

Bayesian Optimization (BO)

- Gaussian Process (GP) most commonly used prior:
  - can model a large class of functions[1]
  - analytically and computationally tractable
- Usual BO algorithms have two steps:
  1. obtain the posterior on f based on prior and observations.
  2. Query point selection rule:

$$x_t = \arg\max_{x \in \mathcal{X}} \quad \phi_t(x)$$

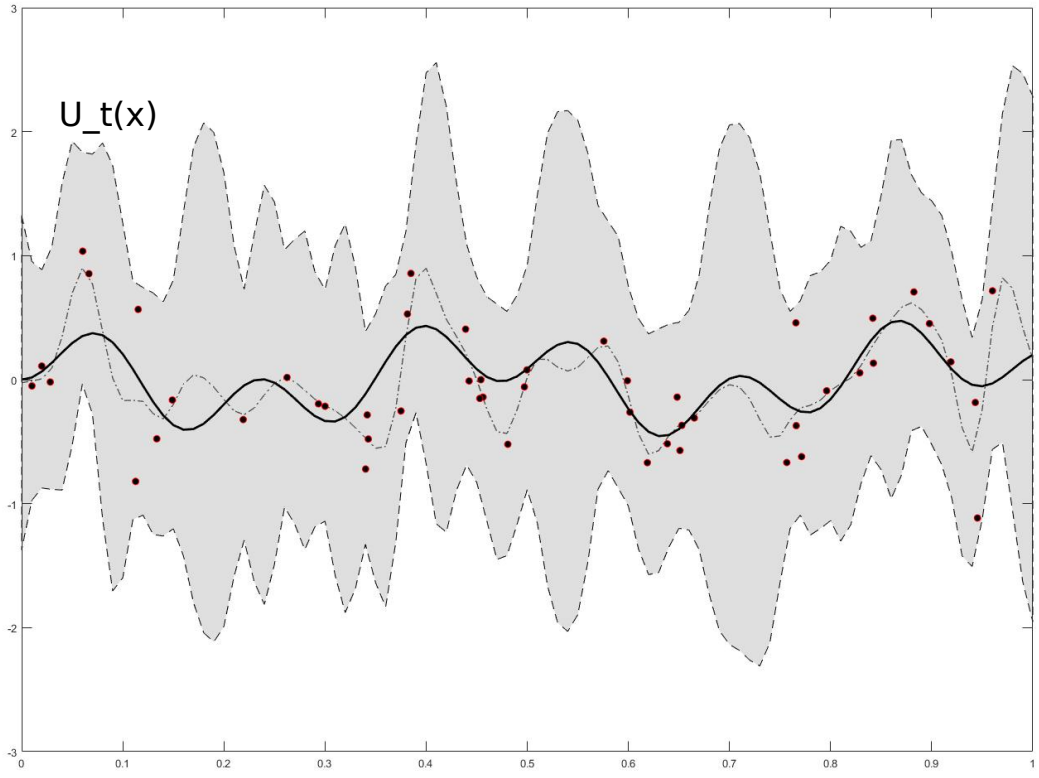
  where $\phi_t(x)$ represents the utility of x.

- examples UCB, EI, PI, Thompson Sampling
- Require maximization of ϕ_t over the continuous space $\mathcal{X}$.

[1] Micchelli et al. (2006). *Universal kernels*. JMLR

GP Confidence Intervals

UCB algorithm: $x_t \in \arg\max_{x \in \mathcal{X}} U_t(x)$



Lipschitz Optimization

- Algorithms adaptively partition the search space $\mathcal{X}$.
- Idea: can discard regions based on observations.
  Example: Suppose $f : [0, 1] \to \mathbb{R}$ is 1-Lipschitz.
  - Let $f(0.2) \in [1, 1.1]$ and $f(0.8) \in [0.4, 0.5]$
  - Then $x^* \notin (0.3, 1]$
- At any time t algorithms divide $\mathcal{X}$ into $\mathcal{O}(t)$ regions.
  - query points selected from $\mathcal{O}(t)$ representative points.
  - no global maximization over continuous space $\mathcal{X}$ required.
- Algorithms such as zooming algorithm[2], and various tree based methods[3]
- Drawback: Lipschitz asumption too strict.

[2] Kleinberg et al.(2013)
[3] Munos (2014)

BO with adaptive discretization

Question: Can we combine

1. the representation power of GPs
2. and the computational simplicity of Lipschitz optimization

Answer: Yes.

- We present two algorithms:
  - A Bayesian version of the *zooming algorithm*
  - A Bayesian tree based algorithm.
- We use techniques from the study of suprema of GPs to design the algorithms.
- BaMSOO[4] only other algorithm which attempts this in a much restricted setting.

[4] Wang et al. (2014)

Some notations

- $GP(0, K)$ induces a metric on $\mathcal{X}$, denoted by $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$

$$d(x_1, x_2) = [K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2)]^{1/2}$$

- $l(x_1, x_2) = \|x_1 - x_2\|$.
- Covariance function K satisfies two assumptions:
  - **A1:** $\forall x, y \in \mathcal{X}$: $d(x, y) \leq g(\|x - y\|)$, for $g : \mathbb{R}^+ \to \mathbb{R}^+$ non-decreasing
  - **A2:** $\exists \delta_0, C_K, \alpha > 0$, such that $g(r) \leq C_K r^\alpha \quad \forall r \leq \delta_0$.
- $\mathcal{K} = \{K : K$ satisfies A1 and A2$\}$.

Algorithm-1: Bayesian Zooming algorithm

- The algorithm maintains a set A_t of points that have been evaluated at least once.
- Based on posterior mean and variance, construct $L_t(x), U_t(x)$ for all $x \in A_t$.
- If $n_t(x) = k$, then we can show that $\sigma_t(x) \leq \sigma/\sqrt{k}$
- To each $x \in A_t$, such that $n_t(x) = k$, we assign a radius r_k
- For each (x, r_k) pair we have a bound $W(r_k)$ on the variation of f in $B(x, r_k)$.
- New evaluation points are selected *optimistically*

Bayesian Zooming algorithm

Algorithm 1: Bayesian Zooming Algorithm

Input : $n > 0$, $(r_k)_{k\geq 0}$, $(W(r_k))_{k\geq 0}$

while $t \leq n$ **do**
  choose $x_t = \arg\max_{x\in A_t} U_t(x) + W(r_{n_t(x)})$
  evaluate $y_t = f(x_t) + \eta_t$
  update posterior $\mu_t(x)$ and $\sigma_t(x)$
  update $n_{t+1}(x_t) \leftarrow n_t(x_t) + 1$
  update $r_{t+1}(x)$
  if $\mathcal{X} \not\subset \cup_{x_i\in A_t} B(x_i, r_{t+1}(x_i))$ **then**
    Add a point $x \in \mathcal{X} \setminus \cup_{x_i\in A_t} B(x_i, r_{t+1}(x_i))$ to A_t, with $r_t(x) = r_0 = diam(\mathcal{X})$.
  end
end

Regret Bounds for zooming algorithm

Theorem-1

With high probability, the cumulative regret incurred by Algorithm- 1 is upper bounded by

$$\mathcal{R}_n \leq \tilde{\mathcal{O}}\big(n^{1-\frac{\alpha}{\tilde{D}+2\alpha}}\big) \tag{1}$$

Here $\tilde{D}$ is a notion of dimension of the near optimal regions of f and $\tilde{D} \leq D$ a.s.

Outline:

- if $n_t(x) \geq k$, then $f(x^*) - f(x) \leq \tilde{\mathcal{O}}(1/\sqrt{k})$
- if $n_t(x), n_t(y) \leq k$, then $l(x, y) \geq r_k$
- *Suboptimal points are widely spaced* $\Rightarrow$ bound them with their packing numbers.

Comparison with existing bounds

- Existing bounds on $\mathcal{R}_n$ have the general form:

$$\mathcal{R}_n \leq \mathcal{O}(\sqrt{n\gamma_n \log n}) \tag{2}$$

- Here γ_n is the maximum *information gain* from n observations

$$\gamma_n = \sup_{S \subset \mathcal{X} : |S| = n} I(y_S; f) \tag{3}$$

- To get explict nontrivial bounds, we need *sublinear* bounds on γ_n for specific kernels.

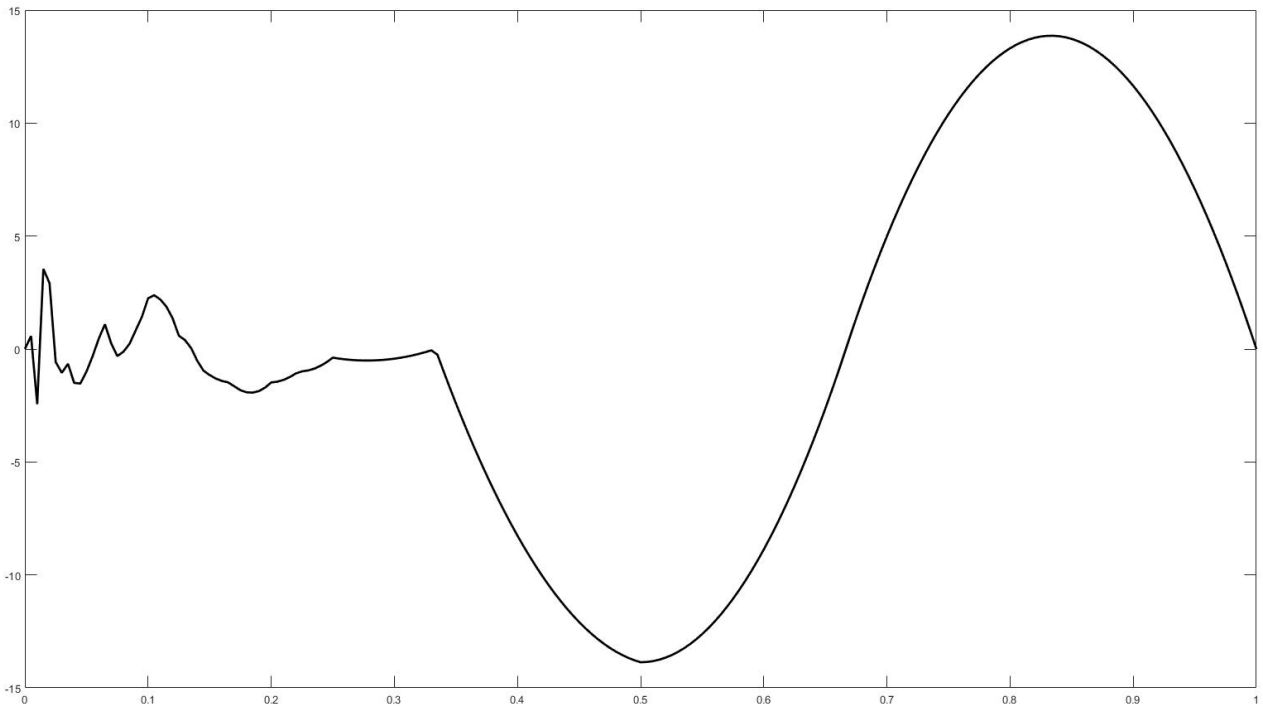
- γ_n: maximum information about f, and not necessarily x^*.

A toy example

Suppose $\mathcal{X} = [0,1]$ and let $f : [0,1] \to \mathbb{R}$ be a sample from:

$$f(x) = \sum_{i=1}^{\infty} a_i X_i \big(\psi(3^i x - 1) - \psi(3^i x - 2)\big)$$

$$\psi(x) = 1 - 4(x - 0.5)^2$$



Improved bounds for Matérn kernels

- Matérn kernels are a widely used in ML. Parameterized by $\nu = m + 1/2$.

$$K(r) = K(0)(1 + p_m(r))e^{-c_1\sqrt{\nu}r}$$

- Our bounds improve on the existing bounds in two ways:
  - For $\nu = 1/2$, we provide the first explicit sublinear bounds on cumulative regret.
  - For all other ν, our bounds are tighter when $D \geq \nu - 1$.

- Most commonly used in ML are $\nu = 3/2$ and $\nu = 5/2$.

Algorithm-2: Tree based algorithm

- The zooming algorithm requires a covering oracle to check whether :

$$\mathcal{X} \not\subset \cup_{x_i \in A_t} B(x_i, r_{t+1}(x_i))$$

holds, and if not return any point from the uncovered region.

- Can be difficult to implement for arbitrary metric spaces.

- Alternative: work with a fixed sequence (or tree) of partitions:
  - Finite subsets $(\mathcal{X}_h)_{h \geq 0}$, where $\mathcal{X}_h = \{x_{h,i} : 1 \leq i \leq 2^h\}$
  - for each $x_{h,i}$, we have a cell

$$\mathcal{X}_{h,i} = \{x \in \mathcal{X} : l(x, x_{h,i}) \leq l(x, x_{h,j}) \quad \forall j \neq i\}$$

$\mathcal{X}_{h,i}$ for a fixed h, partition $\mathcal{X}$.

Regret bounds for Tree based algorithm

Theorem-2

Suppose the tree of partitions has cells of geometrically decaying diameters (in the metric l). Then we have w.h.p.

$$\mathcal{S}_n = f(x^*) - f(x(n)) \leq \tilde{\mathcal{O}}(n^{-\alpha/(\tilde{D}+2\alpha)}) \tag{4}$$

$$\mathcal{R}_n = \sum_{t \leq n} f(x^*) - f(x_t) \leq \tilde{\mathcal{O}}(n^{1-\frac{\alpha}{\tilde{D}+2\alpha}}) \tag{5}$$

Outline:

- if $x_{h,i}$ is evaluated, then $f(x^*) - f(x_{h,i}) \leq 4V_{h-1}$
- points at level h in the tree are separated by some ρ_h.
- number of such points can be bounded by packing numbers.

Comparison with BaMSOO

- Another algorithm which works on a tree of partitions is Bayesian Multi-Scale Optimistic Optimization (BaMSOO)
- Evaluates points at all levels h of the current tree.
- Bound on $\mathcal{S}_n$ of the form $\tilde{\mathcal{O}}(n^{-c/D})$ for some $c > 0$.

Our method has some advantages:

- BaMSOO requires extra assumptions for regret guarantees: doesn't hold for $K(x_1, x_2) = c_1 \exp(-c_2 \|x_1 - x_2\|)$.
- BaMSOO only works with noiseless observations.
- $\mathcal{S}_n$ for BaMSOO is always $\tilde{\mathcal{O}}(n^{-c/D})$. For our algorithm for some GP, $\mathcal{S}_n = \tilde{\mathcal{O}}(e^{-c'n})$.

Summary of Results

- We present two algorithms for Bayesian Optimization, based on ideas from Lipschitz optimization.
- We derive some bounds on the variation of GP samples in d-balls to facilitate the choice of parameters.
- We obtain bounds on cumulative regret in terms of near-optimality dimension:
  - tighten the bounds for Matérn kernels.
  - first explicit sub-linear bounds for exponential kernels.
  - construct a toy-example showing when γ_n based bounds are loose
- Obtain bounds on Simple Regret for second algorithm:
  - Some improvements over BaMSOO

BACKUP SLIDES

Regret bounds for zooming algorithm

With high probability, the following are true:

- If a point x_t is chosen to be evaluated at time t, then we have for $\Delta(x_t) = f(x^*) - f(x_t)$

$$\Delta(x_t) \leq \mathcal{O}\left(\frac{\sqrt{\log n}}{\sqrt{n_t(x_t)}}\right) \tag{6}$$

- If two points x and y have been evaluated no more than k times each must be separated by a distance of r_k

- Let $\rho^i \leq h < \rho^{i+1}$, and for $\Delta_i = (\frac{B'_n}{\rho^{i/2}})$, we define $\mathcal{X}_{\Delta_i} = \{x \in \mathcal{X} : \Delta(x) \leq \Delta_i\}$. Contribution of the points evaluated h times for $h \in [\rho^i, \rho^{i+1}]$ to the cumulative regret:

$$\bar{\mathcal{R}}_i \leq \rho^{i+1}\left(\frac{B'_n}{\rho^{i/2}}\right) M(\mathcal{X}_{\Delta_i}, r_{\rho^i}, l) \tag{7}$$

Extension to agnostic setting

- Assumption: f is an arbitrary function in the RKHS[5] of the kernel K with a knonwn bound (B) on the RKHS norm.
- By reproducing property, Cauchy-Schwarz inequality and assumptions on kernel K:

$$|f(x_1) - f(x_2)| = |\langle f, K(x_1, \cdot)\rangle - \langle f, K(x_2, \cdot)\rangle| \leq Bd(x_1, x_2)$$
$$\leq Bg(\|x_1 - x_2\|)$$

- We can apply the zooming algorithm here, to get a similar regret bound as Eq.(1)
- For Matérn kernels, our bounds are sublinear for all values of ν and D. Existing bounds are sublinear only if $\nu > D(D+1)$.

[5] Reproducing Kernel Hilbert Space

Algorithm-2: Tree based algorithm

- Assume that the cells $\mathcal{X}_{h,i}$ satisfy:
  - $\mathcal{X}_{h,i} \subset B(x_{h,i}, R_h)$
  - $B(x_{h,i}, r_h) \subset \mathcal{X}_{h,i}$
- Select points optimistically from current leaf set $\mathcal{L}_t$. Initially $\mathcal{L}_0 = \{x_{01}\}$= root node
- After selecting point x_{h_t,i_t}, one of two actions:
  - Evaluate: If $n_t(x_{h_t,i_t}) < K_{h_t}$, evaluate f at x_{h_t,i_t}.
  - Expand: If $n_t(x_{h_t,i_t}) = K_{h_t}$, expand node (h_t, i_t).

Tree based algorithm

Algorithm 2: Tree based Algorithm for Bayesian Optimization

Input : $n > 0$, $(\mathcal{X}_h)_{h\geq 0}$, $(V_h)_{h\geq 0}$, $(K_h)_{h\geq 0}$, $\mathcal{L}_0 = \{x_{0,1}\}$

for $t = 1$ *to* n **do**
  choose $x_{h_t,i_t} = \arg\max_{x_i \in \mathcal{L}_t} I_t(x_{h,i}) =$
  $\mu_{t-1}(x_{h,i}) + B_n \sigma_{t-1}(x_{h,i}) + V_h$
  if $n_t(x_{h_t,i_t}) < K_{h_t}$ **then**
    $y_t = f(x_{h_t,i_t}) + \eta_t$
    $n_{t+1}(x_{h_t,i_t}) = n_t(x_{h_t,i_t}) + 1$
    update posterior $\mu_t(x)$ and $\sigma_t(x)$
  else
    $\mathcal{L}_{t+1} = \mathcal{L}_t \setminus \{(h_t, i_t)\}$
    $\mathcal{L}_{t+1} = \mathcal{L}_{t+1} \cup \{(h_t + 1, 2i_t - 1), (h_t + 1, 2i_t)\}$
  end
end

Output: $x(n)$: the deepest expanded node

Regret analysis of Tree based algorithm

With high probability, the following statements are true:

- If a point $x_{h,i}$ is expanded by the algorithm, then we must have $f(x_{h,i}) + 3V_h \geq f(x^*)$ which means that $f(x_{h,i}) \geq f(x^*) - 3V_h$.
- If a point $x_{h,i} \in \mathcal{L}_t$ and $p(x_{h,i}) = x_{h-1,\lfloor (i+1)/2 \rfloor}$, then it must satisfy $f(x_{h,i}) \geq f(p(x_{h,i})) - V_{h-1} \geq f(x^*) - 4V_{h-1}$.
- Thus, at level h the algorithm only selects points from the set $\mathcal{I}_h = \{x \in \mathcal{X}_h : f(x_h) \geq f(x^*) - 4V_{h-1}\}$

Regret analysis of Tree based algorithm

Suppose we select the points according to Algorithm-1. Let us define $H(n)$ in the following way:

$$H(n) = \max\{H : \sum_{h \geq 0}^{H} K_h |\mathcal{I}_h| < n\}$$

Then the point recommended by the algorithm, the simple regret for recommending $x(n)$ will satisfy the following w.h.p.

$$\mathcal{S}_n = f(x^*) - f(x(n)) \leq 3V_{H(n)} \tag{8}$$

Moreover, for any $H > 0$, we have the following high probability bound on cumulative regret:

$$\mathcal{R}_n = \sum_{t \leq n} f(x^*) - f(x_t) \leq \sum_{h=0}^{H} K_h |\mathcal{I}_h| 4V_{h-1} + 4nV_H \tag{9}$$