Bayesian Optimization using Pseudo-Points

Chao Qian\textsuperscript{1}\textsuperscript{*}, Hang Xiong\textsuperscript{2} and Ke Xue\textsuperscript{1}

\textsuperscript{1}National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
\textsuperscript{2}University of Science and Technology of China, Hefei 230027, China

\{qianc, xuek\}@lamda.nju.edu.cn, flybear@mail.ustc.edu.cn

Abstract

Bayesian optimization (BO) is a popular approach for expensive black-box optimization, with applications including parameter tuning, experimental design, and robotics. BO usually models the objective function by a Gaussian process (GP), and iteratively samples the next data point by maximizing an acquisition function. In this paper, we propose a new general framework for BO by generating pseudo-points (i.e., data points whose objective values are not evaluated) to improve the GP model. With the classic acquisition function, i.e., upper confidence bound (UCB), we prove that the cumulative regret can be generally upper bounded. Experiments using UCB and other acquisition functions, i.e., probability of improvement (PI) and expectation of improvement (EI), on synthetic as well as real-world problems clearly show the advantage of generating pseudo-points.

1 Introduction

One often needs to solve an optimization problem: \(x^{*} \in \arg\max_{x \in X} f(x)\), where \(X \subseteq \mathbb{R}^d\) is the solution space, \(f : X \to \mathbb{R}\) is the objective function, and \(x^{*}\) is an optimal solution. Usually, it is assumed that \(f\) has a known mathematical expression, is convex, or cheap to evaluate at least. Increasing evidences, however, show that \(f\) may not satisfy these assumptions, but is an expensive black-box model [Brochu \textit{et al.}, 2010]. That is, \(f\) can be non-convex, or even the closed-form expression of \(f\) is unknown; meanwhile, evaluating \(f\) can be noisy and computationally very expensive.

Expensive black-box optimization is involved in many real-world decision making problems. For example, in machine learning, one has to tune hyper-parameters to maximize the performance of a learning algorithm [Snoek \textit{et al.}, 2012]; in physical experiments, one needs to set proper parameters of the experimental environment to obtain an ideal product [Brochu \textit{et al.}, 2010]. More applications can be found in robotic control [Martinez-Cantin \textit{et al.}, 2007], computer vision [Denil \textit{et al.}, 2012], sensor placing [Garnett \textit{et al.}, 2010], and analog circuit design [Lyu \textit{et al.}, 2018].

BO [Mockus, 1994] has been a type of powerful algorithm to solve expensive black-box optimization problems. The main idea is to build a model, usually by a GP, for the objective \(f\) based on the observation data, and then sample the next data point by maximizing an acquisition function. Many BO algorithms have been proposed, with the goal of reaching the optima using as few objective evaluations as possible.

Most existing works focus on designing effective acquisition functions, e.g., PI [Kushner, 1964], EI [Jones \textit{et al.}, 1998], and UCB [Srinivas \textit{et al.}, 2010]. Recently, Wang \textit{et al.} [2016] proposed the EST function by directly estimating \(x^{*}\), which automatically and adaptively trades off exploration and exploitation in PI and UCB. Another major type of acquisition function is based on information entropy, including entropy search (ES) [Hennig and Schuler, 2012], predictive ES [Hernández-Lobato \textit{et al.}, 2014], max-value ES [Wang and Jegelka, 2017], FITBO [Ru \textit{et al.}, 2018], etc. As BO is a sequential algorithm, some parallelization techniques have been introduced for acceleration, e.g., [Azimi \textit{et al.}, 2010; Desautels \textit{et al.}, 2014; Shah and Ghahramani, 2015; González \textit{et al.}, 2016]. There is also a sequence of works addressing the difficulty of BO for high-dimensional optimization, e.g., [Wang \textit{et al.}, 2013; Kandasamy \textit{et al.}, 2015; Wang \textit{et al.}, 2017; Mutny and Krause, 2018].

For any BO algorithm with a specific acquisition function, the GP model becomes increasingly accurate with the observation data augmenting. However, the number of data points to be evaluated is often limited due to the expensive objective evaluation. In this paper, we propose a general framework for BO by generating pseudo-points to improve the GP model. That is, before maximizing the acquisition function to select the next point in each iteration, some pseudo-points are generated and added to update the GP model. The pseudo-points are neighbors of the observed data points, and take the same function values as the observed ones. Without increasing the evaluation cost, the generation of pseudo-points can reduce the variance of the GP model, while introducing little accuracy loss under the Lipschitz assumption. This framework is briefly called BO-PP.

Theoretically, we study the performance of BO-PP w.r.t. the acquisition function UCB, called UCB-PP. We prove a
Algorithm 1 BO Framework

Input: iteration budget $T$

Process:
1: let $D_0 = \emptyset$;
2: for $t = 1 : T$ do
3: \[ x_t = \arg \max_{x \in X} \text{acq}(x); \]
4: evaluate $f$ at $x_t$ to obtain $y_t$;
5: augment the data $D_t = D_{t-1} \cup \{(x_t, y_t)\}$ and update the GP model
6: end for

The general framework of BO is shown in Algorithm 1. It sequentially optimizes a given objective function $f(x)$ with assumptions on a prior distribution, i.e., a probabilistic model, over $f(x)$. In each iteration, BO selects a point $x$ by maximizing an acquisition function $\text{acq}()$, evaluates its objective value $f(x)$, and updates the prior distribution with the new data point.

2.1 GPs

A GP [Rasmussen and Williams, 2006] is commonly used as the prior distribution, which regards the $f$ value at each data point as a random variable, and assumes that all of them satisfy a joint Gaussian distribution specified by the mean value function $m(\cdot)$ and the covariance function $k(\cdot, \cdot)$. For convenience, $m(\cdot)$ is set to zero. Assume that the objective evaluation is subject to i.i.d. additive Gaussian noise, i.e., $y = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Let $[t_1, t_2, \ldots, t_T]$ denote the set \( t \).

Given an observation data $D_t = \{(x_i, y_i)\}_{i=1}^t$, we can obtain the posterior mean
\[ \mu_t(x) = k_t(x)^T(K_t + \sigma^2I)^{-1}y_{1:t}, \]
and the posterior variance
\[ \sigma_t^2(x) = k(x, x) - k_t(x)^T(K_t + \sigma^2I)^{-1}k_t(x), \]
where $k_t(x) = [k(x, x_{i1})]_{i=1}^t$, $K_t = [k(x_i, x_j)]_{i,j \in [t]}$ and $y_{1:t} = [y_1; y_2; \ldots; y_t]$. For a GP, the log likelihood of observed data $D_t$ is
\begin{equation}
\log \Pr(y_{1:t} | \{x_i\}_{i=1}^t, \Theta) = -(1/2)y_{1:t}^T(K_t + \sigma^2I)^{-1}y_{1:t} - (1/2) \log \det(K_t + \sigma^2I) - (t/2) \log 2\pi,
\end{equation}
where $\Theta$ denote the hyper-parameters of $k(\cdot, \cdot)$, and $\det(\cdot)$ denotes the determinant of a matrix. When updating the GP model in line 5 of Algorithm 1, the hyper-parameters $\Theta$ can be updated by maximizing the log likelihood of the augmented data, or treated to be fully Bayesian.

2.2 Acquisition Functions

The data point to be evaluated in each iteration is selected by maximizing an acquisition function, which needs to trade off exploration, i.e., large posterior variances, and exploitation, i.e., large posterior means. Many acquisition functions have been proposed, and we introduce three typical ones, i.e., PI [Kushner, 1964], EI [Jones et al., 1998] and UCB [Srinivas et al., 2012], which will be examined in this paper.

Let $x^*$ be the best point generated in the first $(t-1)$ iterations, and $Z = (\mu_{t-1}(x) - f(x^*))/\sigma_{t-1}(x)$. Let $\Phi$ and $\phi$ denote the cumulative distribution and probability density functions of standard Gaussian distribution, respectively. PI selects the point by maximizing the probability of improvement, i.e.,
\[ \text{PI}(x) = \Pr(f(x) > f(x^*)) = \Phi(Z). \]

EI selects the data point by maximizing the expectation of improvement, i.e.,
\[ \text{EI}(x) = \begin{cases} (\mu_{t-1}(x) - f(x^*))\Phi(Z) + \sigma_{t-1}(x)\phi(Z) & \text{if } \sigma_{t-1}(x) > 0, \\ 0 & \text{if } \sigma_{t-1}(x) = 0. \end{cases} \]

UCB integrates the posterior mean and variance via a trade-off parameter $\beta_t$, i.e.,
\[ \text{UCB}(x) = \mu_{t-1}(x) + \beta_t^{1/2}\sigma_{t-1}(x), \]
and selects the data point by maximizing this measure.

2.3 Regrets

To evaluate the performance of BO algorithms, regrets are often used. The instantaneous regret $r_i = f(x^*) - f(x_i)$ measures the gap of function values between an optimal solution $x^*$ and the currently selected point $x_i$. The simple regret $S_T = \min_{i \in [T]} r_i$ measures the gap between $x^*$ and the best point found in the first $T$ iterations. The cumulative regret $R_T = \sum_{t=1}^T r_i$ is the sum of instantaneous regrets in the first $T$ iterations. A BO algorithm is said to be no-regret if $\lim_{T \to \infty} R_T/T = 0$.

3 The BO-PP Framework

In BO, a GP is used to characterize the unknown objective function. The posterior variance of a GP describes the uncertainty about the unknown objective, while the posterior mean provides a closed form of the unknown objective. As the observation data augments, the posterior variance decreases and the posterior mean gets close to the unknown objective, making the GP express the unknown objective better. Thus, a straightforward way to improve the GP model is collecting more data points, which is, however, impractical, because the objective evaluation is expensive. In this section, we propose a general framework BO-PP by generating pseudo-points to improve the GP model.

As shown in Eq. (2), the posterior variance of $f$ does not depend on the objective values, and will be decreased by adding new data points. As shown in Eq. (1), the posterior mean of $f$ can be regarded as a linear combination of the observed objective values, and will be influenced by the error on
Algorithm 2 BO-PP Framework

**Input:** iteration budget $T$

**Parameter:** $\{l_i\}_{i=1}^{T-1}, \{\tau_i\}_{i=0}^{T-1}$

**Process:**

1. let $D_0 = \emptyset$ and $l_0 = 0$;
2. for $t = 1 : T$ do
3. generate $l_{t-1}$ pseudo-points $\{(x'_i, y'_i)\}_{i=1}^{l_{t-1}}$;
4. re-compute $\mu_{t-1}$ and $\sigma_{t-1}$ by $D_{t-1} \cup \{(x'_i, y'_i)\}_{i=1}^{l_{t-1}}$;
5. $x_t = \arg \max_{x \in X} \text{acq}(x)$;
6. evaluate $f$ at $x_t$ to obtain $y_t$;
7. augment the data $D_t = D_{t-1} \cup \{(x_t, y_t)\}$ and update the GP model
8. end for

where each pseudo-point in the $t$-th iteration has distance $\tau_{t-1}$ to some observed data point in $D_{t-1}$, and takes the same objective value as the observed one.

4 Theoretical Analysis

In this section, we theoretically analyze the performance of BO-PP w.r.t. the acquisition function UCB, called UCB-PP. Specifically, we prove that the cumulative regret $R_T$ of UCB-PP can be generally upper bounded.

We first give some notations that will be used in the following analysis. Let $\mu_t$ and $\sigma_t$ denote the posterior mean and variance after obtaining $D_t$; let $\tilde{\mu}_t$ and $\tilde{\sigma}_t$ denote the posterior mean and variance after adding pseudo-points $\{(x'_i, y'_i)\}_{i=1}^{l_i}$ into $D_t$; let $\tilde{\mu}_t$ and $\tilde{\sigma}_t$ denote the posterior mean and variance after adding pseudo-points with true observed objective values, i.e., $\{(x'_i, y'_i)\}_{i=1}^{l_i}$, where $y'_i = f(x'_i) + \epsilon'_i$ with $\epsilon'_i \sim N(0, \sigma^2)$. Some notations about pseudo-points: $y'_{i,l} = (y'_1; y'_2; \ldots; y'_l)$, $y'_{i,l} = [y'_1; y'_2; \ldots; y'_l]$, $K'_t(x) = [k(x, x'_1) \ldots k(x, x'_l)]_{l \times l}$, $K'_t,l = [k(x, x'_1) \ldots k(x, x'_l)]_{l \times l}$, $s(x) = K'_t,l(K'_t + \sigma^2 I)^{-1}k_t(x) - k'_t(x)$, $M = (K'_t,l - K'_t,l)(K'_t + \sigma^2 I)^{-1}K'_t,l + \sigma^2 I)^{-1}$. For convenience of analysis, assume $k(x, x) = 1$.

Let $A$ be a finite subset of $X$. $\mathcal{A}$ denote true objective values (which are actually random variables satisfying the posterior Gaussian distribution over the true objective values), and $y_A(x)$ denote the noisy observations. Let $PP$ denote all generated pseudo-points, and $\tilde{y}_PP$ denote their selected objective values. Note that $y_{PP}$ are random variables, as they are actually the noisy observations of the objective values of $PP$’s neighbor observed points. Let $l_{\mathcal{A}} = \max_{x \in \mathcal{A}} I(y_A; \tilde{y}_{PP})$ and $l_{\mathcal{A}}$ denote the mutual information.

Theorem 1 gives an upper bound of UCB-PP on the cumulative regret $R_T$. As the analysis of UCB in [Srinivas et al., 2012], Assumption 1 is required, implying

$$
\Pr(\forall x, x': |f(x) - f(x')| \leq L \|x - x'\|_1) \geq 1 - dae^{-(L/b)^2}. \tag{6}
$$

**Assumption 1.** Suppose the kernel $k(\cdot, \cdot)$ satisfies the following probability bound on the derivatives of $f$: for some constants $a, b > 0$, $\forall j \in [d] : \Pr(\sup_{x \in X} |f(x) - f(x_j)| \leq L) \leq a e^{-(L/b)^2}$.

**Theorem 1.** Let $X \subset [0, r]^d$, $d \in (0, 1)$, and set $\beta_t$ in Eq. (5) as $\beta_t = 2 \log^{2}(2\pi t^2/(3\delta)) + 2 d \log(t^2 d b r \sqrt{\log(4d a / \delta)})$. Running UCB-PP for $T$ iterations, it holds that

$$
\Pr\left(R_T \leq \sqrt{C T \beta_T} \gamma_T^2 + 2 + 2 \sum_{i=1}^{T} \Delta_m(l_{i-1}, \tau_{i-1})\right) \geq 1 - \delta, \tag{7}
$$

where $C = 8/\log(1 + \sigma^{-2})$, and $\Delta_m(l, \tau) = l^2 \sqrt{1 + \sigma^{-2}} \left(b d r \sqrt{\log(4d a / \delta)} / \sigma + 2 \sqrt{\log(4 \sum_{j=0}^{l-1} l_j \delta)} \right)$.

Lemma 1 bounds the error on the posterior mean led by the incorrect objective values of pseudo-points, which will be used in the proof of Theorem 1. Its proof is provided in the supplementary material due to space limitation.

**Lemma 1.** After obtaining $D_t$ in UCB-PP, the difference on the posterior mean by adding pseudo-points, i.e., $\{(x'_i, y'_i)\}_{i=1}^{l_i}$, and that with true observed objective values, i.e., $\{(x'_i, y'_i)\}_{i=1}^{l_i}$, is $\tilde{\mu}_t(x) - \tilde{\mu}_t(x) = -p(x)^T M \{y'_{i,l} - y'_{i,l}\}$. Furthermore, it holds that

$$
\Pr(\forall 0 \leq t \leq T - 1, \forall x \in X : |\tilde{\mu}_t(x) - \tilde{\mu}_t(x)| \leq \Delta_m(L, l_t, \tau_t)) \geq 1 - dae^{-(L/b)^2} - \delta / 4,
$$

where $\Delta_m(L, l_t, \tau_t) = l^2 \sqrt{1 + \sigma^{-2}} \left(b d r \sqrt{\log(4d a / \delta)} / \sigma + 2 \sqrt{\log(4 \sum_{j=0}^{l-1} l_j \delta)} \right)$.
The proof of Theorem 1 is inspired by that of Theorem 2 in [Srinivas et al., 2012], which gives an upper bound of UCB on the cumulative regret $R_T$. Their proof intuition is mainly that the instantaneous regret $r_t$ can be upper bounded by the width of confidence interval of $f(x_t)$, relating to the posterior variance. The generation of pseudo-points will introduce another quantity into the upper bound on $r_t$, characterized by the error on the posterior mean in Lemma 1.

**Proof of Theorem 1.** According to Assumption 1 and $\beta_t = 2 \log(2\pi^2/(d \ell^2 T))$, where $L = b/\sqrt{\log(4da/\delta)}$, we can apply Lemma 5.7 in [Srinivas et al., 2012] to derive that

$$\Pr(\forall t \geq 1 : |f(x^*) - \check{\mu}_{t-1}(x^*)|) \leq \beta_t^{1/2} \check{\sigma}_{t-1}(x^*) + 1/\ell^2 \geq 1 - \delta/2,$$  

(8)

where $|x^*|$ denotes the discretized data point closest to $x^*$ in the $t$-th iteration. Note that $\Delta_m(L, \tau_1)$ is just $\Delta_m(L, \ell_1, \tau_1)$ with $L = b/\sqrt{\log(4da/\delta)}$ in Lemma 1. By the definition of $r_t$, we have, for all $t \geq 1$,

$$r_t = f(x^*) - f(x_t) \leq \beta_t^{1/2} \check{\sigma}_{t-1}(x^*) + \check{\mu}_{t-1}(x^*) - f(x_t) + 1/\ell^2 \leq \beta_t^{1/2} \check{\sigma}_{t-1}(x^*) + \check{\mu}_{t-1}(x^*) - f(x_t) + 1/\ell^2 + \Delta_m(L, \ell_1, \tau_1) = \beta_t^{1/2} \check{\sigma}_{t-1}(x^*) + \check{\mu}_{t-1}(x^*) - f(x_t) + 1/\ell^2 + \Delta_m(L, \ell_1, \tau_1) \leq \beta_t^{1/2} \check{\sigma}_{t-1}(x_t) + \check{\mu}_{t-1}(x_t) - f(x_t) + 1/\ell^2 + \Delta_m(L, \ell_1, \tau_1) \leq \beta_t^{1/2} \check{\sigma}_{t-1}(x_t) + \check{\mu}_{t-1}(x_t) - f(x_t) + 1/\ell^2 + 2\Delta_m(L, \ell_1, \tau_1) \leq 2\beta_t^{1/2} \check{\sigma}_{t-1}(x_t) + 1/\ell^2 + 2\Delta_m(L, \ell_1, \tau_1),$$

where the first inequality holds with probability at least $1 - \delta/2$ by Eq. (8), the second and fourth inequalities hold with probability at least $1 - dae^{-\ell(b/2)^2} - \delta/4 = 1 - \delta/2$ by Lemma 1, the equality holds because the posterior variance in Eq. (2) does not depend on the objective values, leading to $\forall x : \check{\sigma}_{t-1}(x) = \check{\sigma}_{t-1}(x)$, the third inequality holds because $x_t$ is selected by maximizing $\check{\mu}_{t-1}(x) + \beta_t^{1/2} \check{\sigma}_{t-1}(x)$ in Eq. (5), and the last inequality holds with probability at least $1 - \delta/4$ by Lemma 5.5 in [Srinivas et al., 2012]. Note that to prove Lemma 5.7 in [Srinivas et al., 2012] and Lemma 1, Assumption 1, i.e., Eq. (6), is both used; thus, the probability $iae^{-\ell(b/2)^2} = \delta/4$ has been repeated. By the union bound, we have

$$\Pr(\forall t \geq 1 : r_t \leq 2\beta_t^{1/2} \check{\sigma}_{t-1}(x_t) + 1/\ell^2 + 2\Delta_m(l_{t-1}, \tau_{t-1})) \geq 1 - \delta/2 - \delta/\ell^2 - \delta/4 = 1 - \delta,$$

implying

$$\Pr\left(R_T = \sum_{t=1}^{T} r_t \leq \sum_{t=1}^{T} (2\beta_t^{1/2} \check{\sigma}_{t-1}(x_t) + 1/\ell^2 + 2\Delta_m(l_{t-1}, \tau_{t-1})) \right) \geq 1 - \delta.$$  

By the Cauchy–Schwarz inequality, $C = 8/\log(1 + \sigma^2)$ and $\forall t \leq T : \beta_t \leq \beta_T$, we have

$$\sum_{t=1}^{T} 2\beta_t^{1/2} \check{\sigma}_{t-1}(x_t) \leq \sqrt{2T} \sum_{t=1}^{T} \beta_t^{1/2} \check{\sigma}_{t-1}^2(x_t) \leq \sqrt{\frac{CT\beta_T}{2}} \sum_{t=1}^{T} \log(1 + \sigma^2 \check{\sigma}_{t-1}^2(x_t)).$$

Let $PP_t$ denote the pseudo-points generated in the $t$-th iteration, and $y_{PP_t}$ denote their selected objective values. Let $H(.)$ denote the entropy. We have

$$\frac{1}{2} \sum_{t=1}^{T} \log(1 + \sigma^2 \check{\sigma}_{t-1}^2(x_t)) \leq \frac{1}{2} \sum_{t=1}^{T} \log(1 + \sigma^2 \check{\sigma}_{t-1}^2(x_t)) + \frac{1}{2} \log(\det(2\pi e\sigma^2 I)) \leq \frac{1}{2} \sum_{t=1}^{T} \log(2\pi e(\sigma^2 + \check{\sigma}_{t-1}^2(x_t))) = H(y_1) + y_{PP_t} + H(y_2 | y_1, y_{PP_t}) + \cdots + H(y_T | y_{T-1}, y_{PP_T}) = H(y_1 | y_{PP}) + H(y_2 | y_1, y_{PP}) + \cdots + H(y_T | y_{T-1}, y_{PP}) = H(y_1) + y_{PP} + H(y_2 | y_1, y_{PP}) + \cdots + H(y_T | y_{T-1}, y_{PP})$$

where the first equality holds because $f(x)$ is subject to additive Gaussian noise $\mathcal{N}(0, \sigma^2)$. Thus,

$$\frac{1}{2} \sum_{t=1}^{T} \log(1 + \sigma^2 \check{\sigma}_{t-1}^2(x_t)) \leq H(y_1) + y_{PP} + H(y_2 | y_1, y_{PP}) + \cdots + H(y_T | y_{T-1}, y_{PP}) = \gamma_T.$$  

Considering $\sum_{t=1}^{T} 1/\ell^2 = \pi^2/6 < 2$, Eq. (7) holds. Thus, the theorem holds.

Adding pseudo-points is to improve the GP model when the number of observed data points is not large. After UCB-PP runs many iterations, there are already enough observed points, and thus pseudo-points are not needed. That is, UCB-PP will only add pseudo-points in a finite number of iterations, denoted by $T_0$. This implies that $\forall t \geq T_0, \ell_t = 0$, leading to $\Delta_m(l_t, \tau_t) = 0$. Thus, lim$_{T \to +\infty}$ $R_T/T = 0$, implying that UCB-PP is no-regret.

Under the same assumption, it has been proved [Srinivas et al., 2012] that the cumulative regret $R_T$ of UCB satisfies

$$\Pr(R_T \leq \sqrt{\frac{CT\beta_T\gamma_T}{2}} + 2) \geq 1 - \delta,$$  

(9)

where $\gamma_T = \max_{A:|A|=T} I(y_A; f_A)$, and the other parameters have the same meaning as that in Theorem 1. Without generating pseudo-points, $\forall t \geq 0 : l_t = 0 \land I(y_A; y_{PP}) = 0$, and thus, $\gamma_T = \gamma_T \land \Delta_m(l_t, \tau_t) = 0$, implying that Eq. (7) specializes to Eq. (9). Thus, we have:

**Remark 1.** Our bound on $R_T$ of UCB-PP is a generalization of the bound on $R_T$ of UCB in [Srinivas et al., 2012].

As $\gamma_T \leq \gamma_T$, the comparison between Eqs. (7) and (9) suggests that the generation of pseudo-points can be helpful if the negative influence of introducing the error on the posterior mean, i.e., introducing the term $\Delta_m(l_t, \tau_t)$, can be compensated by the positive influence of reducing the posterior variance, i.e., introducing the term $I(y_A; y_{PP})$. 
In this section, we empirically compare BO-PP with BO. Three common acquisition functions, i.e., UCB, PI and EI, are used. The ARD squared exponential kernel is employed, whose hyper-parameters are tuned by maximum likelihood estimation (MLE), and the acquisition function is maximized via the DIRECT algorithm [Jones et al., 1993]. To alleviate the “cold start” issue, each algorithm starts with five random initial points. To compare BO-PP with BO on each problem, we repeat their running 20 times independently and report the average results; in each running, BO-PP and BO use the same five random initial points. The noise level is set to $\sigma^2 = 0.0001$, and the iteration budget is set to 100.

In the $(t+1)$-th iteration of BO-PP, for each point in $D_t$, one pseudo-point is generated by randomly sampling within its distance $\tau_t$ and taking the same function value; thus, $l_t = \lfloor D_t \rfloor$. To control the error of objective values with pseudo-points increasing, $\tau_t$ is set to $r \tau_0/(dl_t)$, which decreases with $l_t$. Note that $r$ corresponds to the width of each dimension of the search domain. $\tau_0$ is set to a small value. We will use 0.01, 0.001 and 0.0001 to explore its influence, and the corresponding algorithms are denoted as BO-PP01, BO-PP001 and BO-PP0001, respectively.

We use four common synthetic benchmark functions: Dropwave, Griewank, Hart6 and Rastrigin, whose dimensions are 2, 2, 6 and 2, respectively. Their search domains are scaled to $[-1,1]^d$. As the minima are known, the simple regret $S_T$ is used as the metric. We also employ four real-world optimization problems, widely used in BO experiments [Springenberg et al., 2016; Wang and Jegelka, 2017; Ru et al., 2018]. The first is to tune the hyper-parameters, i.e., box constraint $C \subseteq [0.001, 1000]$ and kernel scale $l \in [0.0001, 1]$, of SVM for classification on the data set Wine quality (1,599 #inst, 11 #feat). The second is to tune the hyper-parameters of 1-hidden-layer neural network (NN) for this task. The NN is trained by backpropagation, and the hyper-parameters are the number of neurons $n \in [1,100]$ and the learning rate $l \in [0.000001, 1]$. The last two problems are to tune the hyper-parameters of 1-hidden-layer NN for classification on Breast cancer (699 #inst, 9 #feat) and regression on Boston housing (506 #inst, 13 #feat), respectively. The NN is trained by Levenberg-Marquardt optimization, and there are four hyper-parameters: $n \in [1,100]$, the damping factor $\mu \in [0.000001, 100]$, the $\mu$-decrease and $\mu$-increase factors $\mu_{\text{dec}}, \mu_{\text{inc}} \in [0.01, 1], \mu_{\text{inc}} \in [1.01, 20]$. All data sets are randomly split into training/validation/test sets with ratio 0.7/0.2/0.1, and the performance on validation sets is used as the objective $f$. For classification, $f$ is the classification accuracy; for regression, $f$ equals 20 minus the regression L2-loss.

For UCB, $\beta_t$ in Eq. (5) is set to $2 \log(t^{3/2}+2\pi^2/3\delta)$ where $\delta = 0.1$, as suggested in [Brochu et al., 2010; Srinivas et al., 2012]. For PI and EI, the best observed function value by far is used as $f(x^*)$ in Eqs. (3) and (4). The results are summarized in Table 1. We can observe that UCB-PP0001
is always better than UCB, and UCB-PP01/UCB-PP001 surpasses UCB in most cases, disclosing that the performance of UCB-PP is not very sensitive to the distance $\tau_t$. Also, PI-PP and EI-PP perform better than PI and EI, respectively, in most cases, showing the applicability of generating pseudo-points.

Furthermore, we plot the curves of the simple regret $S_T$ or the objective $f$ over iterations for each algorithm on each problem. Figure 1 shows the curves of UCB-PP and UCB on real-world problems. It can be observed that on each problem, there is at least one curve of UCB-PP almost always above that of UCB, implying that UCB-PP can consistently outperform UCB during the running process. The other five figures, showing similar observations, are provided in the supplementary material due to space limitation.

To examine the robustness of BO-PP against kernels, we use the Gaussian kernel with hyper-parameters tuned by MLE. We compare UCB-PP with UCB on real-world optimization problems. Figure 2 shows that UCB-PP can be better than UCB except UCB-PP01/UCB-PP001 on $SVM_{wine}$.

6 Conclusion

In this paper, we propose a general framework BO-PP by generating pseudo-points to improve the GP model of BO. BO-PP can be implemented with any acquisition function. Equipped with UCB, we prove that the cumulative regret of BO-PP can be well bounded. This bound generalizes the well-known bound of UCB. Experiments with UCB, PI and EI on synthetic as well as real-world optimization problems show the superior performance of BO-PP over BO. It is expected that the generation of pseudo-points can be helpful for more BO algorithms. Note that the dimensionality of problems tested in our experiments is low. Thus, studying the effectiveness of BO-PP on high-dimensional optimization problems is an interesting topic.

It is also noted that the added pseudo-points take the same function values with their neighbor observed data points, requiring the function to vary smoothly locally. Thus, it is interesting to study strategies of improving BO when the function can fluctuate widely in the future.
References


Appendix

7.1 Detailed Proofs

This part provides the proof of Lemma 1, which is omitted in Section 4 of our original paper due to space limitation. We first give two lemmas which will be used in the proof of Lemma 1.

Lemma 2. After obtaining \( D_t \) in UCB-PP, the reduction on the posterior variance by adding pseudo-points is

\[
\Delta_0(x, t, \tau_t) = \sigma_t^2(x) - \delta_t^2(x) = p(x)^T M p(x).
\]

Proof. By Eq. (2) and \( k(x, x) = 1 \), we have

\[
\delta_t^2(x) = 1 - \left[ k_t(x); k'_t(x) \right]^T \left( \begin{bmatrix} K_t & K'_t \end{bmatrix} + \sigma^2 I \right)^{-1} \left[ k_t(x); k'_t(x) \right]
\]

\[
= 1 - k_t(x)^T (K_t + \sigma^2 I)^{-1} k_t(x) - p(x)^T M p(x)
\]

where the second equality is derived by the inverse of a block matrix, and the third one holds by Eq. (2). Thus, the lemma holds.

Lemma 3 is extracted from Lemma 5.1 in [Srinivas et al., 2012].

Lemma 3. Suppose \( X \) is a random variable satisfying the Gaussian distribution \( \mathcal{N}(\mu, \sigma^2) \). Then, it holds that

\[
\Pr(|X - \mu| > c \sigma) \leq e^{-c^2/2},
\]

where \( c > 0 \).

Proof of Lemma 1. By Eq. (1), we have

\[
\hat{\mu}_t(x) = [k_t(x); k'_t(x)]^T \left( \begin{bmatrix} K_t & K'_t \end{bmatrix} + \sigma^2 I \right)^{-1} \left[ k_t(x); k'_t(x) \right]
\]

\[
= k_t(x)^T (K_t + \sigma^2 I)^{-1} y_t + p(x)^T M \mu_{\{\hat{\mu}_t(x_1); \hat{\mu}_t(x_2); \cdots; \hat{\mu}_t(x_t)\}} - y_t,
\]

where the second equality is derived by the inverse of a block matrix, and the third one holds by Eq. (1). Similarly, we have

\[
\hat{\mu}_t(x) = [k_t(x); k'_t(x)]^T \left( \begin{bmatrix} K_t & K'_t \end{bmatrix} + \sigma^2 I \right)^{-1} \left[ k_t(x); k'_t(x) \right]
\]

\[
= k_t(x)^T (K_t + \sigma^2 I)^{-1} y_t + p(x)^T M \mu_{\{\hat{\mu}_t(x_1); \hat{\mu}_t(x_2); \cdots; \hat{\mu}_t(x_t)\}} - y_t,
\]

Thus, it holds that \( \hat{\mu}_t(x) = -p(x)^T M (\hat{y}_t - t, y_t) \).

Next, we examine the upper bound on \( |p(x)^T \mathcal{M}(\hat{y}_t - t, y_t)| \),

\[
|p(x)^T \mathcal{M}(\hat{y}_t - t, y_t)| \leq \sum_{i=1}^{l_t} |p(x)^T \mathcal{M}(\hat{y}_t - t, y_t)| \leq \sum_{i=1}^{l_t} \|p(x)^T \mathcal{M}(\hat{y}_t - t, y_t)\|_1
\]

where \( \|p(x)^T \mathcal{M}\|_i \) denotes the i-th element of \( p(x)^T \mathcal{M} \). According to the procedure of Algorithm 2, the pseudo-point \( x'_j \) has distance \( \tau_t \) with some observed data point and takes the same function value. Assume that the corresponding observed data point for \( x'_j \) is \( x_j \), where \( j \in [t] \), implying \( y'_j = y_j \). Thus, we have

\[
|y'_j - y_j| = |y_j - y'_j| = \|f(x_j) + \epsilon_j - (f(x'_j) + \epsilon'_{j})\|
\]

\[
\leq \|f(x_j) - f(x'_j)\| + \epsilon_j - \epsilon'_{j},
\]

where \( \epsilon_j, \epsilon'_{j} \sim \mathcal{N}(0, \sigma^2) \), and the second equality holds because \( f(x_j) \) and \( f(x'_j) \) are subject to additive Gaussian noise \( \mathcal{N}(0, \sigma^2) \). According to Assumption 1, we have

\[
\Pr(\forall t \geq 0, \forall i \in [l_t]: |f(x_j) - f(x'_j)| \leq L \|x_j - x'_j\|_1 = Ld \tau_t)
\]

\[
\geq 1 - dae^{-(L/b)^2}.
\]

As \( \epsilon_j - \epsilon'_j \sim \mathcal{N}(0, 2\sigma^2) \), by Lemma 3, we have

\[
\Pr(|\epsilon_j - \epsilon'_j| \leq 2\sigma \sqrt{\frac{L d}{b}} \leq \frac{\delta}{\sqrt{t}} = \frac{\delta}{\sqrt{t}})
\]

\[
\geq 1 - \delta e^{-(L/b)^2} - \frac{\delta}{\sqrt{t}}.
\]

Thus, with probability at least \( 1 - \frac{\delta}{4} \), it holds that

\[
\Pr(\forall t \leq T - 1, \forall i \in [l_t]: |y_j - y_j'| \leq Ld \tau_t + 2\sigma \sqrt{\frac{L d}{b}} \leq \frac{\delta}{\sqrt{t}})
\]

\[
\geq 1 - \frac{\delta}{4}.
\]

Next we prove an upper bound on \( \sum_{i=1}^{l_t} |p(x)^T \mathcal{M}(y'_t ; y_t)| \).

\[
|\lim_{l_t \to +\infty} \sum_{i=1}^{l_t} |p(x)^T \mathcal{M}(y'_t ; y_t)| \leq \sum_{i=1}^{l_t} |p(x)^T \mathcal{M}(y'_t ; y_t)|
\]

where \( p(x)^T \mathcal{M} \) denotes the j-th element of \( p(x)^T \mathcal{M} \). According to the Cauchy interlacing inequality, leading to \( \lambda_k(M^{-1}) \geq \lambda_k(M^{-1}) \geq \cdots \geq \lambda_k(M^{-1}) \geq \lambda_k(M^{-1}) \geq \lambda_k(M^{-1}) \geq \lambda_k(M^{-1}) \geq \sigma^2 \). Combining Eqs. (11), (12) and (13), we have

\[
\sum_{i=1}^{l_t} |p(x)^T \mathcal{M}(y'_t ; y_t)| \leq \sum_{i=1}^{l_t} \|p(x)^T \mathcal{M}(y'_t ; y_t)\|_1
\]

\[
\leq \sum_{i=1}^{l_t} \|p(x)^T \mathcal{M}(y'_t ; y_t)\|_1
\]

\[
\leq \sum_{i=1}^{l_t} \|p(x)^T \mathcal{M}(y'_t ; y_t)\|_1
\]

\[
\leq \sum_{i=1}^{l_t} \|p(x)^T \mathcal{M}(y'_t ; y_t)\|_1
\]
Applying Eq. (14) to Eq. (10), we have

\[
Pr \left( \forall 0 \leq t \leq T - 1, \forall x \in X : | p(x)^T M( \hat{y}_{i_{t}, i_{t}} - y_{i_{t}}') | \right) 
\leq \left( Ld\tau + 2\sigma \sqrt{\log 4 \sum_{t=0}^{T-1} l_t} \right) \sqrt{1 + \sigma^2 \sigma^{-2} l_t^2} 
\geq 1 - \frac{d}{a} e^{-\left( \frac{L}{b} \right)^2} - \frac{\delta}{4}.
\]

Thus, the lemma holds. \( \Box \)

7.2 Detailed Experimental Results

This part provides some experimental results, which are omitted in Section 5 of our original paper due to space limitation.

Figure 3 plots the curves of the simple regret \(S_T\) over iterations for UCB-PP and UCB on synthetic benchmark functions.

Figure 4 plots the curves of the objective \(f\) over iterations for PI-PP and PI on real-world optimization problems.

Figure 5 plots the curves of the simple regret \(S_T\) over iterations for PI-PP and PI on synthetic benchmark functions.

Figure 6 plots the curves of the objective \(f\) over iterations for EI-PP and EI on real-world optimization problems.

Figure 7 plots the curves of the simple regret \(S_T\) over iterations for EI-PP and EI on synthetic benchmark functions.
Figure 3: The results (mean±(1/4)std.) of UCB-PP and UCB on synthetic benchmark functions. $S_T$: the smaller, the better.

Figure 4: The results (mean±(1/4)std.) of PI-PP and PI on real-world optimization problems. $f$: the larger, the better.

Figure 5: The results (mean±(1/4)std.) of PI-PP and PI on synthetic benchmark functions. $S_T$: the smaller, the better.
Figure 6: The results (mean±(1/4)std.) of EI-PP and EI on real-world optimization problems. $f$: the larger, the better.

Figure 7: The results (mean±(1/4)std.) of EI-PP and EI on synthetic benchmark functions. $S_T$: the smaller, the better.