Combining Bayesian Optimization and Lipschitz Optimization

Mohamed Osama Ahmed  
moahmed@cs.ubc.ca

Sharan Vaswani  
sharanv@cs.ubc.ca

Mark Schmidt  
schmidtm@cs.ubc.ca

University of British Columbia

Abstract

Bayesian optimization and Lipschitz optimization have developed alternative techniques for optimizing black-box functions. They each exploit a different form of prior about the function. In this work, we explore strategies to combine these techniques for better global optimization. In particular, we propose ways to use the Lipschitz continuity assumption within traditional BO algorithms, which we call Lipschitz Bayesian optimization (LBO). This approach does not increase the asymptotic runtime and in some cases drastically improves the performance (while in the worst case the performance is similar). Indeed, in a particular setting, we prove that using the Lipschitz information yields the same or a better bound on the regret compared to using Bayesian optimization on its own. Moreover, we propose a simple heuristics to estimate the Lipschitz constant, and prove that a growing estimate of the Lipschitz constant is in some sense “harmless”. Our experiments on 15 datasets with 4 acquisition functions show that in the worst case LBO performs similar to the underlying BO method while in some cases it performs substantially better. Thompson sampling in particular saw drastic improvements (as the Lipschitz information corrected for its well-known “over-exploration” phenomenon) and its LBO variant often outperformed other acquisition functions.

1 Introduction

Bayesian optimization (BO) has a long history and has been used in a variety of fields (see Shahriari et al. 2016), with recent interest from the machine learning community in the context of automatic hyperparameter tuning (Snoek et al. 2012, Golovin et al. 2017). BO is an example of a global black-box optimization algorithm (Hendrix et al. 2010, Jones et al. 1998, Pintér 1991, Rios and Sahinidis 2013) which optimizes an unknown function that may not have nice properties such as convexity. In the typical setting, we assume that we only have access to a black box that evaluates the function and that it is expensive to do these evaluations. The objective is to find a global optimum of the unknown function with the minimum number of function evaluations.

The global optimization of a real-valued function is impossible unless we make assumptions about the structure of the unknown function. Lipschitz continuity (that the function can’t change arbitrarily fast as we change the inputs) is one of the weakest assumptions under which optimizing an unknown function is still possible. Lipschitz optimization (Piyavskii 1972, Shubert 1972, LO) exploits knowledge of the Lipschitz constant of the function (a bound on the amount that it can change) to prune the search space in order to locate the optimum. In contrast, Bayesian optimization makes a stronger assumption that the unknown function belongs to a known model class (typically a class of smooth functions), the most common being a Gaussian process (GP) generated using a Gaussian or Matérn kernel (Stein 2012). We review LO and BO in Section 2.

Under their own specific sets of additional assumptions, both BO (Bull 2011, Theorem 5) and LO (Malherbe and Vayatis 2017) can be shown to be exponentially faster than random search strategies. If the underlying function is close to satisfying the stronger BO assumptions, then typically BO is able to optimize functions faster than LO. However, when these assumptions are not reasonable, BO may converge
slower than simply trying random values (Li et al. 2016; Ahmed et al. 2016). On the other hand, LO makes minimal assumptions (not even requiring differentiability) and simply prunes away values of the parameters that are not compatible with the Lipschitz condition and thus cannot be solutions. This is useful in speeding up simple algorithms like random search. Given a new function to optimize, it is typically not clear which of these strategies will perform better.

In this paper, we propose to combine BO and LO to exploit the advantages of both methods. We call this Lipschitz Bayesian Optimization (LBO). Specifically, in Section 3, we design mixed acquisition functions that use Lipschitz continuity in conjunction with existing BO algorithms. We also address the issue of providing a “harmless” estimate of the Lipschitz constant (see Section 2.3), which is an important practical issue for any LO method. Our experiments (Section 4) indicate that in some settings the addition of estimated Lipschitz information leads to a huge improvement over standard BO methods. This is particularly true for Thompson sampling, which often outperforms other standard acquisition functions when augmented with Lipschitz information. This seems to be because the estimated Lipschitz continuity seems to correct for the well-known problem of over-exploration (Shahriari et al. 2014). Further, our experiments indicate that it does not hurt to use the Lipschitz information since even in the worst case it does not change the runtime or the performance of the method.

2 Background

We consider the problem of maximizing a real-valued function \( f \) with parameters \( x \) over a compact set \( \mathcal{X} \). We assume that on iteration \( t \), an algorithm chooses a point \( x_t \in \mathcal{X} \) and then receives the corresponding function value \( f(x_t) \). Typically, our goal is to find the largest possible \( f(x) \) across iterations. We describe two approaches for solving this problem, namely BO and LO, in detail below.

2.1 Bayesian Optimization

BO methods are typically based on Gaussian processes (GPs), since they have appealing universal consistency properties over compact sets and admit a closed-form posterior distribution (Rasmussen and Williams 2006). BO methods typically assume a smooth GP prior on the unknown function, and use the observed function evaluations to compute a posterior distribution over the possible function values at any point \( x \). At iteration \( t \), given the previously selected points \( \{x_1, x_2, \ldots, x_{t-1}\} \) and their corresponding observations \( y_t = [y_1, y_2, \ldots, y_{t-1}] \), the algorithm uses an acquisition function (based on the GP posterior) to select the next point to evaluate. The value of the acquisition function at a point characterizes the importance of evaluating that point in order to maximize \( f \). To determine \( x_t \), we usually maximize this acquisition function over all \( x \) using an auxiliary optimization procedure (typically we can only approximately solve this maximization).

We now formalize the above high-level procedure. We assume that \( f \) follows a \( GP(0, k(x, x')) \) distribution where \( k(x, x') \) is a kernel function which quantifies the similarity between points \( x \) and \( x' \). Throughout this paper, we use the Matérn kernel for which \( k(x, x') = \frac{1}{\sigma^2} \exp\left(-\sqrt{5}r + \frac{5}{2}r^2\right) \) where \( r = \sum_{j=1}^d \frac{(x_j - x'_j)^2}{\ell_j} \). Here the hyper-parameter \( \ell_j \) is referred to as the length-scale for dimension \( j \) and dictates the extent of smoothness we assume about the function \( f \) in direction \( j \). The hyper-parameter \( \sigma \) quantifies the amount of noise we expect in the function values.

We denote the maximum value of the function until iteration \( t \) as \( y^*_t \) and the set \( \{1, 2, \ldots, t\} \) as \( \mathcal{T} \). Let \( k_t(x) = [k(x, x_1), k(x, x_2), \ldots, k(x, x_t)] \) and let us denote the \( t \times t \) kernel matrix as \( K \) (so \( K_{ij} = k(x_i, x_j) \) for all \( i, j \in \mathcal{T} \)). Given the function evaluations (observations), the posterior distribution at point \( x \) after \( t \) iterations is given as \( \mathcal{P}[f_t(x)] \sim N(\mu_t(x), \sigma_t(x)) \). Here, the mean and standard deviation of the function at \( x \) are given as:

\[
\mu_t(x) = k_t(x)^T (K + \sigma^2 I_t)^{-1} y_t,
\]

\[
\sigma_t(x) = k_t(x) - k_t(x)^T (K + \sigma^2 I_t)^{-1} k_t(x).
\] (1)

As alluded to earlier, an acquisition function uses the above posterior distribution in order to select the next point to evaluate the function at. A number of acquisition functions have been proposed in the literature, with the most popular ones: (UCB) (Srinivas et al. 2010), Thompson sampling (TS) (Thompson 1933), expected improvement (EI) (Mockus 1975), probability of improvement (PI) (Kushner 1964), and entropy search (Villemonteix et al. 2009; Hennig and Schuler 2012; Hernández-Lobato et al. 2014). In this work, we focus on four simple widely-used acquisition functions: UCB, TS, EI, and PI. However, we expect that our conclusions would apply to other acquisition functions. For brevity, when defining the acquisition functions, we drop the \((t-1)\) subscripts from \( \mu_{t-1}(x) \), \( \sigma_{t-1}(x) \), and \( y^*_{t-1} \).
Lipschitz optimization uses this Lipschitz inequality for a constant $L$ where we have defined $z(x)$ we change $x$ assume that the function cannot change too quickly as a specific family of functions, in LO we simply as-

As opposed to assuming that the function comes from a real world process, a suitable value of $L$ is typically dictated by physical limitations of the process. However, in practice we often do not know $L$ and thus need to estimate it. A simple way to obtain an under-estimate $L_t^{ib}$ of $L$ at iteration $t$ is to use the maximum value that satisfies the Lipschitz inequality across all pairs of points, 

$$L_t^{ib} = \max_{i,j \in [t], x_i \neq x_j} \left\{ \frac{|f(x_i) - f(x_j)|}{||x_i - x_j||_2} \right\}. \tag{7}$$
Note that this estimate monotonically increases as we see more examples, but that it may be far smaller than the true $L$ value. A common variation is to sample several points on a grid (or randomly) to use in the estimate above. Unfortunately, without knowing the Lipschitz constant we do not know how fine this grid should be so in general this may still significantly under-estimate the true quantity.

A reasonable property of any estimate of $L$ that we use is that it is “harmless” in the sense of Ahmed et al (2016). Specifically, the choice of $L$ should not make the algorithm converge to the global optimum at a slower speed than random guessing (in the worst case). If we have an over-estimate for the minimum possible value of $L$, then the LO algorithm is harmless as it can only prune values that cannot improve the objective function (although if we over-estimate it by too much then it may not prune much of the space). However, the common under-estimates of $L$ discussed in the previous paragraph are not harmless since they may prune the global optima.

We propose a simple solution to the problem that LO is not harmless if we don’t have prior knowledge about $L$: we use a growing estimate of $L$. The danger in using a growing strategy is that if we grow $L$ too slowly then the algorithm may not be harmless. However, in the appendix we show that LO is “harmless” for most reasonable strategies for growing $L$. This result is not prescriptive in the sense that it does not suggest a practical strategy for growing $L$ (since it depends on the true $L$), but this result shows that even for enormous values of $L$ that an estimate would have to be growing exceedingly slowly in order for it to not be harmless (exponentially-slow in the minimum value of $L$, the dimensionality, and the desired accuracy).

In our experiments we simply use $L^\text{ub}_t = \kappa t \cdot L^\text{lb}_t$, the under-estimator multiplied by the (growing) iteration number and a constant $\kappa$ (a tunable hyper-parameter). In Section 4 we observe that this choice of $L^\text{ub}_t$ with $\kappa = 10$ consistently works well across 14 datasets with 4 different acquisition functions.

3 Lipschitz Bayesian optimization

In this section, we show how simple changes to the standard acquisition functions used in BO allow us to incorporate the Lipschitz inequality bounds. We call this Lipschitz Bayesian Optimization (LBO). LBO prevents BO from considering values of $x^t$ that cannot be global maxima (assuming we have over-estimated $L$) and also restricts the range of $f(x_t)$ values considered in the acquisition function to those that are consistent with the Lipschitz inequalities. Figure 1 illustrates the key features of BO, LO, and LBO. It is important to note that the Lipschitz constant $L$ has a different interpretation than the length-scale $\ell$ of the GP. The constant $L$ specifies an absolute maximum rate of change for the function, while $\ell$ specifies how quickly a parameterized distance between pairs of points changes the GP. We also note that the computational complexity of using the Lipschitz inequalities is $O(n^2)$ which is cheaper than the $O(n^3)$ cost of (exact) inference in the GP.

We can use the Lipschitz bounds to restrict the limits of the unknown function value for computing the improvement. The upper bound $U$ will always be $f^*(x)$, while the lower bound $L$ will depend on the relative value of $y^*$. In particular, we have the following two cases:

$$L = \begin{cases} g^*, & \text{if } y^* \in (f^t(x), f^u(x)) \\ f^u(x), & \text{if } y^* \in (f^u(x), \infty) \end{cases}$$

The second case represents points that cannot improve over the current best value (that are “rejected” by the Lipschitz inequalities).

**Truncated-PI:** We can define a similar variant for the PI acquisition function as:

$$TPI(x) = \Phi(z(x, L)) - \Phi(z(x, U))$$

**Truncated-EI:** Using the above bounds, the truncated expected improvement for point $x$ is given by:

$$TEI(x) = -\sigma(x) \cdot z(x, y^*) \left[ \Phi(z(x, L)) - \Phi(z(x, U)) \right] + \sigma(x) \cdot [\phi(z(x, L)) - \phi(z(x, U))]$$

Note that removing the Lipschitz bounds corresponds to using $f^t(x) = -\infty$ and $f^u(x) = \infty$, and in this case we recover the usual PI and EI methods in Equations 3 and 4, respectively.

**Truncated-UCB:** The same strategy can be applied to UCB as follows:

$$TUCB(x) = \min \left\{ \mu(x) + \beta_t^{1/2} \sigma(x), f^u(x) \right\}$$

**Accept-Reject:** An alternative strategy to incorporate the Lipschitz bounds is to use an accept-reject based mixed acquisition function. This approach uses the Lipschitz bounds as a sanity-check to accept or reject the value provided by the original acquisition function, similar to LO methods. Formally, if $g(x)$ is the value of the original acquisition function (e.g. $g(x) = UCB(x)$ or $g(x) = f^t(x)$ for TS), then the mixed acquisition function $\tilde{g}(x)$ is given as follows:

$$\tilde{g}(x) = \begin{cases} g(x), & \text{if } g(x) \in [f^t(x), f^u(x)] \text{ (Accept)} \\ -\infty, & \text{otherwise (Reject)} \end{cases}$$

We refer to the accept-reject based mixed acquisition functions as AR-UCB and AR-TS, respectively. Note
that the accept-reject method is quite generic and can be used with any acquisition function that has values on the same scale as that of the function. When using an estimate of $L$ it is possible that a good point could be rejected because the estimate of $L$ is too small, but using a growing estimate ensures that such points can again be selected on later iterations.

3.1 Regret bound for AR-UCB

In this section, we show that under reasonable assumptions, AR-UCB is provably “harmless”, in the sense that it retains the good theoretical properties of GP-UCB. We prove the following theorem under the following assumptions:

1. The GP is correctly specified and with infinite observations, the posterior distribution will collapse to the “true” function $f$.

2. The noise in the observations $\sigma$ is small enough for the Lipschitz bounds in Equations 6 to hold.

3. The Lipschitz constant $L$ is known or has been over-estimated using the techniques described in Section 2.3.

Assumption 1 is a common assumption made for providing theoretical results for GP-UCB (Srinivas et al, 2010). Under these assumptions, we obtain the following theorem (proved in Appendix B):

**Theorem 1.** Let $D$ be a finite decision space and $\sigma$ be the standard deviation of the noise in the observations. Let $\pi_i$ be a positive scalar such that $\sum_i \pi_i^{-1} = 1$ and $\delta \in (0,1)$. If we use the AR-UCB algorithm with $\beta_t^{1/2} = 2\log((|D|/\pi_t/\delta)$ assuming that the above conditions 1-3 hold, then the expected cumulative regret $R(T)$ can be bounded as follows:

$$R(T) \leq (8/\log(1+\sigma^{-2}) \beta_T \gamma_T \sqrt{T}$$

Here, $\gamma_T$ refers to the information gain for the selected points and depends on the kernel being used. For the squared exponential kernel, we obtain the following specific bound:

$$R(T) \leq (8/\log(1+\sigma^{-2}) \beta_T (\log(T))^{d+1} \sqrt{T}$$

The $\gamma_T$ term can also be bounded for the Matérn kernel following [Srinivas et al (2010)]. The above theorem shows that under reasonable assumptions, using the Lipschitz bounds in conjunction with GP-UCB cannot result in worse regret. We empirically show that if $L$ is over-estimated, then AR-UCB matches the performance of GP-UCB in the worst case.

Note that the above theorem assumes that the GP is correctly specified with the correct hyper-parameters. It also assumes that we are able to specify the correct value of the trade-off parameter $\beta_t^{1/2}$. These assumptions are not guaranteed to hold in practice and this may result in worse performance of the GP-UCB algorithm. In such cases, our experiments show that using the Lipschitz bounds can lead to better empirical performance than the original GP-UCB.

### 4 Experiments

**Datasets:** We perform an extensive experimental evaluation and present results on twelve synthetic datasets and three real-world tasks. For the synthetic experiments, we use the standard global-optimization benchmarks namely the Branin, Camel, Goldstein Price, Hartmann (2 variants), Michalwicz (3 variants) and Rosenbrock (4 variants). The closed form and
domain for each of these functions is given in Jamil and Yang (2013). As examples of real-world tasks, we consider tuning the parameters for a \textit{robot-pushing} simulation (2 variants) (Wang and Jegelka, 2017) and tuning the hyper-parameters for logistic regression (Wu et al, 2017). For the robot pushing example, our aim is to find a good pre-image (Kaelbling and Lozano-Pérez, 2017) in order for the robot to push the object to a pre-specified goal location. We follow the experimental protocol from Wang and Jegelka (2017) and use the negative of the distance to the goal location as the black-box function to maximize. We consider tuning the robot position $r_x, r_y,$ and duration of the push $t$ for the 3D case. We also tune the angle of the push $\theta_r$ to make it a 4 dimensional problem. For the hyper-parameter tuning task, we consider tuning the strength of the $\ell_2$ regularization (in the range $[10^{-7}, 0.9]$), the learning rate for stochastic gradient descent (in the range $[10^{-7}, 0.05]$) and the number of passes over the data (in the range $[2, 15]$). The black-box function is the negative loss on the test set (using a train/test split of 80%/20%) for the MNIST dataset.

**Experimental Setup:** For Bayesian optimization, we use a Gaussian Process prior with the Matérn kernel (with a different length scale for each dimension). We modified the publically available BO package \textit{pybo} of Hoffman and Shahriari (2014) to construct the mixed acquisition functions. All the prior hyper-parameters were set and updated across iterations according to the open-source Spearmint package\footnote{https://github.com/hips/spearmint}. In order to make the optimization invariant to the scale of the function values, similar to Spearmint, we standardize the function values; after each iteration, we center the observed function values by subtracting their mean and dividing by their standard deviation. We then fit a GP to these rescaled function values and correct for our Lipschitz constant estimate by dividing it by the standard deviation. We use DIRECT (Jones et al, 1993) in order to optimize the acquisition function in each iteration. This is one of the standard choices in current works on BO (Eric et al, 2008; Martinez-Cantin et al, 2007; Mahendran et al, 2012), but we expect that Lipschitz information could improve the performance under other choices of the acquisition function optimization approach such as discretization (Snoek et al, 2012), adaptive grids (Bardenet and Kégl, 2010), and other gradient-based methods (Hutter et al, 2011; Lizotte et al, 2012). In order to ensure that Bayesian optimization does not get stuck in sub-optimal maxima (either because of the auxiliary optimization or a “bad” set of hyper-parameters), on every fourth iteration of BO (or LBO) we choose a random point to evaluate rather than optimizing the acquisition function. This makes the optimization procedure “harmless” in the sense that BO (or LBO) will not perform worse than random search (Ahmed et al, 2016). This has become common in recent BO methods such as [Bull (2011); Hutter et al (2011); and Falkner et al (2017)], and to make the comparison fair we add this “exploration” step to all methods. Note that in the case of LBO we may need to reject random points until we find one satisfying the Lipschitz inequalities (this does not require evaluating the function). In practice, we found that both the standardization and iterations of random exploration are essential for good performance\footnote{Note that we verified that our baseline version of BO performs better than or equal to Spearmint across benchmark problems.}

All our results are averaged over 10 independent runs, and each of our figures plots the mean and standard deviation of the absolute error (compared to the global optimum) versus the number of function evaluations. For functions evaluated on log scale, we show the $10^{\text{th}}$ and $90^{\text{th}}$ quantiles.

**Algorithms compared:** We compare the performance of Random search, BO, and LBO methods (using both estimated and \textit{True} $L$) for the EI, PI, UCB and TS acquisition functions. The \textit{True} $L$ was estimated offline using a large number of random points. For UCB, we set the trade-off parameter $\beta$ according to Kandasamy et al (2017). For EI and PI, we use Lipschitz bounds to truncate the range of function values for calculating the improvement and use the LBO variants TEI and TPI respectively. For UCB and TS, we use the accept-reject strategy and evaluate the LBO variants AR-UCB and AR-TS respectively. In addition to these, we use random exploration as another baseline. We chose the hyper-parameter $\kappa$ (that controls the extent of over-estimating the Lipschitz constant) on the Rosenbrock-4D function and use the best value of $\kappa$ for all the other datasets and acquisition functions for both BO and LBO. In particular, we set $\kappa = 10$.

**Results:** To make the results easier to read, we divide the results into the following groups:

1. LBO provides huge improvements over BO shown in Figure 2. Overall, this represents 21% of all the test cases.

2. LBO provides improvements over BO shown in Figure 3(a). Overall, this represents 9% of all the test cases.

3. LBO performs similar to BO shown in 3(b). Overall, this represents 60% of all the test cases.
4. LBO performs slightly worse than BO shown in Figure 3(a). Overall, this represents 10% of all the test cases.

A comparison of the performance across different acquisition functions (for both BO and LBO) on some of the functions is shown in Figure 4 where we also show an example of UCB where $\beta$ is misspecified. The plots for all functions and methods are available in Appendix C. From these experiments, we can observe:

- LBO can potentially lead to large gains in performance across acquisition functions and datasets, particularly for TS.

- Across datasets, we observe that the gains for EI are relatively small, they are occasionally large for PI and UCB and tend to be consistently large for TS. This can be explained as follows: using EI results in under-exploration of the search space, a fact that has been consistently observed and even theoretically proven by Qin et al (2017). As a result of this, BO does not tend to explore “bad” regions when using EI which results in smaller gains from LBO (on the other hand, it may under-explore).

- TS suffers from exactly the opposite problem: it results in high variance leading to over-exploration of the search space and poor performance. This can be observed in Figures 2(a), 2(b) and 2(c) where the performance of TS is near random. This has also been observed and noted by Shahriari et al (2016). For the discrete multi-armed bandit case, Chapelle and Li (2011) multiply the obtained variance estimate by a small number to discourage over-exploration and show that it leads to better results. LBO offers a more principled way of obtaining this same effect and consequently results in making TS more competitive with the other acquisition functions.

- The only functions where LBO slightly hurts are Rosenbrock-4D and Goldstein with UCB and PI.

- For Michalwicz-5D (Figure 4(a)), we see that there is no gain for EI, PI, or UCB. However, the gain is huge for TS functions. In fact, even though TS is the worst performing acquisition function on this dataset, its LBO variant AR-TS gives the best performance across all methods. This demonstrates the possible gain that can be obtained from using mixed acquisition functions.

- We observe a similar trend in Figures 4(b) where LBO improves TS from near-random performance to being competitive with the best performing methods (while it does not adversely affect the methods performing well).

- For the cases where BO performs slightly better than LBO, we notice that the True estimate of $L$ provides comparable performance to BO, so the problem can be narrowed down to finding a good estimate of $L$.

- Figure 4(c) shows examples where LBO saves BO with UCB when the parameter $\beta$ is chosen too large ($\beta = 10^{16}$). In this case BO performs near random, but using LBO leads to better performance than random search.

In any case, our experiments indicate that LBO methods rarely hurt the performance of the original acquisition function. Since they have minimal computational or memory requirements and are simple to implement, these experiments support using use the Lipschitz bounds.

5 Related work

The Lipschitz condition has been used with BO under different contexts in two previous works (González et al, 2016; Sui et al, 2015). The aim of Sui et al (2015) is to design a “safe” BO algorithm. They assume knowledge of the true Lipschitz constant and exploit Lipschitz continuity to construct a safety threshold in order to construct a “safe” region of the parameter space. This is different than our goal of improving the performance of existing BO methods, and also different in that we estimate the Lipschitz constant as we run the algorithm. On the other hand, González et al (2016) used Lipschitz continuity to model interactions between a batch of points chosen simultaneously in every iteration of BO (referred to as “Batch” Bayesian optimization). This contrasts with our work where we are aiming to improve the performance of existing sequential algorithms (it is possible that our ideas could be used in their framework).

6 Discussion

In this paper, we have proposed simple ways to combine Lipschitz inequalities with some of the most common BO methods. Our experiments show that this often gives a performance gain, and in the worst case it performs similar to a standard BO method. Although we have focused on four of the simplest acquisition functions, it seems that these inequalities could be used within other acquisition functions. Further, we expect that the Lipschitz inequalities could also be
Figure 2: Examples of functions where LBO provides huge improvements over BO for the different acquisition functions. The figure also shows the performance of random search and LBO using the True Lipschitz constant.

Figure 3: Examples of functions where LBO provides some improvement over BO (case a), LBO performs similar to BO (case b), and BO performs slightly better than LBO (case c).

Figure 4: (a, b) Examples of functions where LBO boosts the performance of BO with TS (Better seen in color). (c) Example where LBO outperforms BO with UCB when the $\beta$ parameter is too large ($\beta = 10^{16}$).
used in other settings like BO with constraints (Gelbart et al, 2014; Hernández-Lobato et al, 2016; Gardner et al, 2014). BO methods based on other model classes like neural networks (Snoek et al, 2015) or random forests (Hutter et al, 2011), and methods that evaluate more than one $x_t$ at a time (Ginsbourger et al, 2010; Wang et al, 2016). Finally, there has been recent interest in first-order Bayesian optimization methods (Ahmed et al, 2016; Wu et al, 2017). If the gradient is Lipschitz continuous then it is possible to use the descent lemma (Bertsekas, 2016) to obtain Lipschitz bounds that depend on both function values and gradients.
References

Ahmed MO, Shahriari B, Schmidt M (2016) Do we need harmless Bayesian optimization and first-order Bayesian optimization? NIPS Workshop on Bayesian Optimization

Bardenet R, Kégl B (2010) Surrogating the surrogate: accelerating gaussian-process-based global optimization with a mixture cross-entropy algorithm. In: International Conference on Machine Learning (ICML), Omnipress, pp 55–62

Bertsekas DP (2016) Nonlinear Programming, 3rd edn. MIT

Bull AD (2011) Convergence rates of efficient global optimization algorithms. Journal of Machine Learning Research 12(Oct):2879–2904

Bunin GA, François G (2016) Lipschitz constants in experimental optimization. arXiv preprint arXiv:160307847

Chapelle O, Li L (2011) An empirical evaluation of thompson sampling. In: Advances in Neural Information Processing Systems (NIPS), pp 2249–2257

Eric B, Freitas ND, Ghosh A (2008) Active preference learning with discrete choice data. In: Advances in Neural Information Processing Systems (NIPS), pp 409–416

Falkner S, Klein A, Hutter F (2017) Combining hyperband and bayesian optimization. In: NIPS Workshop on Bayesian Optimization

Gardner JR, Kusner MJ, Xu ZE, Weinberger KQ, Cunningham JP (2014) Bayesian optimization with inequality constraints. In: International Conference on Machine Learning (ICML), pp 937–945

Gelbart MA, Snoek J, Adams RP (2014) Bayesian optimization with unknown constraints. arXiv preprint arXiv:14035607

Ginsbourger D, Le Riche R, Carraro L (2010) Kriging is well-suited to parallelize optimization. In: Computational intelligence in expensive optimisation problems, Springer, pp 131–162

Golovin D, Solnik B, Moitra S, Kochanski G, Karro J, Sculley D (2017) Google vizier: A service for black-box optimization. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 1487–1495

González J, Dai Z, Hennig P, Lawrence N (2016) Batch Bayesian optimization via local penalization. In: International Conference on Artificial Intelligence and Statistics (AISTATS), pp 648–657

Hendrix EM, Boglárka G, et al (2010) Introduction to nonlinear and global optimization. Springer

Hennig P, Schuler CJ (2012) Entropy search for information-efficient global optimization. Journal of Machine Learning Research 13(Jun):1809–1837

Hernández-Lobato JM, Hoffman MW, Ghahramani Z (2014) Predictive entropy search for efficient global optimization of black-box functions. In: Advances in Neural Information Processing Systems (NIPS), pp 918–926

Hernández-Lobato JM, Gelbart MA, Adams RP, Hoffman MW, Ghahramani Z (2016) A general framework for constrained Bayesian optimization using information-based search. Journal of Machine Learning Research 17(1):5549–5601

Hoffman MW, Shahriari B (2014) Modular mechanisms for Bayesian optimization. In: NIPS Workshop on Bayesian Optimization, pp 1–5

Hutter F, Hoos HH, Leyton-Brown K (2011) Sequential model-based optimization for general algorithm configuration. In: International Conference on Learning and Intelligent Optimization, Springer, pp 507–523

Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation 4(2):150–194

Jones DR, Perttunen CD, Stuckman BE (1993) Lipschitzian optimization without the lipschitz constant. Journal of Optimization Theory and Applications 79(1):157–181

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. Journal of Global optimization 13(4):455–492

Kaelbling LP, Lozano-Pérez T (2017) Pre-image backchaining in belief space for mobile manipulation. In: Robotics Research, Springer, pp 383–400

Kandasamy K, Krishnamurthy A, Schneider J, Poczos B (2017) Asynchronous parallel Bayesian optimisation via thompson sampling. arXiv preprint arXiv:170509236

Kushner HJ (1964) A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. Journal of Basic Engineering 86(1):97–106

Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A (2016) Efficient hyperparameter optimization and infinitely many armed bandits. arXiv preprint arXiv:160306560

Lizotte DJ, Greiner R, Schuurmans D (2012) An experimental methodology for response surface optimization methods. Journal of Global Optimization 53(4):699–736
A  Proof for Lipschitz constant estimation

Let $L$ be the minimum possible Lipschitz constant for the function $f$, and let $x^*$ be a globally-optimal solution. Here we consider with the number of iterations we need before we have a point $x_t$ satisfying

$$f(x_t) - f(x^*) \leq \varepsilon,$$

(11)

for some small $\varepsilon$.

By the Lipschitz inequality we have

$$|f(x) - f(x^*)| \leq L \|x - x^*\|,$$

for any $x$. If we assume that $x^*$ is at least $\Omega(\varepsilon/L)$-distance away from the boundary, then this inequality implies that there exists a hyper-sphere around $x^*$ of size $\Omega((\varepsilon/L)^d)$ where all $x$ values satisfy (11). Using that $\mathcal{X}$ is compact (and thus its total volume is bounded), if we choose a random $x$ for some small $\varepsilon$, then by the Lipschitz inequality we have

$$\Pr[\|x - x^*\| \leq \Omega((\varepsilon/L)^d)] = \frac{1}{\mathcal{X}}.$$  

Unfortunately, if we use an estimate $\hat{L}$ of $L$ instead of an $L$ satisfying the Lipschitz inequality, we could reject an approximate solution. However, if $\hat{L}$ grows with $t$ then eventually it is sufficiently large that we will not reject an approximate solution (unless we already have an $\varepsilon$-optimal solution). Thus, a crude bound on the expected number of iterations before we find a solution with accuracy $\varepsilon$ is given by $O((L/\varepsilon)^d + T)$, where $T$ is the first iteration $t$ beyond which we always have $\hat{L} \geq L$. Thus, if we choose the sequence $\hat{L}_t$ such that $T = O((L/\varepsilon)^d)$, then LO is harmless as it requires the same expected number of iterations as random guessing. A simple example of a sequence of $\hat{L}$ values satisfying this property would be to choose $\hat{L}_t = tL(\varepsilon/L)^d$, which grows extremely-slowly (for small $\varepsilon$ and non-trivial $d$ or $L$). Larger sequences would imply a smaller $T$ and hence also would be harmless.

B  Regret Bound

**Theorem 2.** Let $\mathcal{D}$ be a finite decision space and $\sigma$ be the standard deviation of the noise in the observations. Let $\pi_t$ be a positive scalar such that $\sum_t \pi_t = 1$ and $\delta \in (0, 1)$. If we use the AR-UCB algorithm with $\beta_t^{1/2} = 2 \log(|\mathcal{D}|\pi_t/\delta)$ assuming that the above conditions 1-3 hold, then the expected cumulative regret $R(T)$ can be bounded as follows:

$$R(T) \leq (8/\log(1+\sigma^{-2})) \beta_T \gamma_T \sqrt{T}$$

Here, $\gamma_T$ refers to the information gain for the selected points and depends on the kernel being used. For the squared exponential kernel, we obtain the following specific bound:

$$R(T) \leq (8/\log(1+\sigma^{-2})) \beta_T (\log(T))^{d+1} \sqrt{T}$$

**Proof.**

By definition of Lipschitz bounds and assuming we know the true Lipschitz constant $L$, at iteration $t$, for all $x$,

$$f_t^{\dagger}(x) \leq f(x) \leq f_{t-1}^{\dagger}(x)$$

We now use the following lemma from Srinivas et al. (2010):

4If $x^*$ is near the boundary, then the area is still $\Omega((\varepsilon/2L)^d)$ and the argument is unchanged.

5Instead of “number of iterations $t$ to reach $\varepsilon$”, we could equivalently state the result in terms “expected error at iteration $t^*$ (simple regret) by inverting the relationship between $t$ and $\varepsilon$. This would give an expected error on iteration $t$ of $O(L/t^{1/d})$. 


Let us now consider the term \( \beta^t_1 = 2 \log(|D|/\pi_1) \delta \) where \( \delta \in (0, 1) \). Then, for all \( x \in D \) and \( t \geq 1 \), with probability \( 1 - \delta \),

\[
|f(x) - \mu_{t-1}(x)| \leq \beta^t_1/2 \sigma_{t-1}(x) \tag{13}
\]

For the point \( x_t \) selected at round \( t \), the following relation holds because of the Accept-Reject condition:

\[
f_{t-1}^t(x_t) \leq \mu_{t-1}(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t) \leq f_{t-1}^u(x_t) \tag{14}
\]

The following holds because of the definition of the UCB rule:

\[
\mu_{t-1}(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t) \geq \mu_t(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t) \tag{15}
\]

From Equations 12 and 13,

\[
f(x_t) \leq \min\{f_{t-1}^u(x_t), \mu_{t-1}(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t)\} \tag{16}
\]

From Equations 13 and 14

\[
\mu_{t-1}(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t) \leq \min\{f(x_t) + 2\beta^t_1/2 \sigma_{t-1}(x_t), f_{t-1}^u(x_t)\} \tag{17}
\]

Let \( r_t \) be the instantaneous regret in round \( t \). Then,

\[
r_t = f(x_t) - f(x_t)
\]

\[
\leq \min\{f_{t-1}^u(x_t), \mu_{t-1}(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t)\} - f(x_t)
\]

\[
\leq \min\{f_{t-1}^u(x_t), \mu_{t-1}(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t)\} - f(x_t)
\]

\[
= \min\{f_{t-1}^u(x_t) - f(x_t), \mu_{t-1}(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t) - f(x_t)\}
\]

\[
\leq \mu_{t-1}(x_t) + \beta^t_1/2 \sigma_{t-1}(x_t) - f(x_t)
\]

\[
\leq \min\{f(x_t) + 2\beta^t_1/2 \sigma_{t-1}(x_t), f_{t-1}^u(x_t)\} - f(x_t)
\]

\[
= \min\{2\beta^t_1/2 \sigma_{t-1}(x_t), f_{t-1}^u(x_t) - f(x_t)\}
\]

\[
\Rightarrow r_t \leq \min\{2\beta^t_1/2 \sigma_{t-1}(x_t), f_{t-1}^u(x_t) - f_{t-1}(x_t)\}
\]

Let us now consider the term \( f_{t-1}^u(x_t) - f_{t-1}^t(x_t) \)

\[
f_{t-1}^u(x_t) - f_{t-1}^t(x_t) = \min_{i \in [t-1]} \{ f(x_i) + L||x_t - x_i||_2 \} - \max_{i \in [t-1]} \{ f(x_i) - L||x_t - x_i||_2 \} \tag{By Equation 6}
\]

\[
= \min_{i \in [t-1]} \{ f(x_i) + L||x_t - x_i||_2 \} + \max_{i \in [t-1]} \{ -f(x_i) + L||x_t - x_i||_2 \}
\]

\[
\leq \min_{i \in [t-1]} \{ f(x_i) + L||x_t - x_i||_2 - f(x_i) + L||x_t - x_i||_2 \}
\]

\[
\Rightarrow f_{t-1}^u(x_t) - f_{t-1}^t(x_t) \leq 2L \min_{i \in [t-1]} \{ ||x_t - x_i||_2 \}
\]

From the above equations,

\[
r_t \leq \min \left\{ 2\beta^t_1/2 \sigma_{t-1}(x_t), 2L \min_{i \in [t-1]} \{ ||x_t - x_i||_2 \} \right\}
\]
Let $R(T)$ be the cumulative regret after $T$ rounds.

$$R(T) = \sum_{t=1}^{T} r_t \leq \sum_{t=1}^{T} \left[ \min \left\{ 2\beta_t^{1/2}\sigma_{t-1}(x_t), 2L \min_{i \in [t-1]} \{||x_t - x_i||_2\} \right\} \right]$$

$$R(T) \leq \min \left\{ \sum_{t=1}^{T} \beta_t^{1/2}\sigma_{t-1}(x_t), 2L \sum_{t=1}^{T} \min_{i \in [t-1]} \{||x_t - x_i||_2\} \right\} \quad (\min\{\sum_{i} a_i\} \geq \sum_{i} \min\{a_i\})$$

We now bound the term $\sum_{t=1}^{T} \beta_t^{1/2}\sigma_{t-1}(x_t)$ using the lemma in Srinivas et al. (2010) which we restate next:

**Lemma 2** (Lemma 5.4 in Srinivas et al. (2010)). Choosing $\beta_t^{1/2} = 2 \log(|D|/\pi_t/\delta)$, 

$$\sum_{t=1}^{T} \beta_t^{1/2}\sigma_{t-1}(x_t) \leq C_1 \gamma_T \sqrt{T}$$

where $C_1 = (8/\log(1 + \sigma^{-2})) \beta_T$. Here $\gamma_T$ refers to the information gain for the selected points.

Using the above lemma, we obtain the following bound:

$$R(T) \leq \min \left\{ C_1 \gamma_T \sqrt{T}, 2L \sum_{t=1}^{T} \min_{i \in [t-1]} \{||x_t - x_i||_2\} \right\}$$

$$\Rightarrow R(T) \leq (8/\log(1 + \sigma^{-2})) \beta_T \gamma_T \sqrt{T}$$

\[\square\]

### C Additional Experimental Results

Below we show the results of all the experiments for all the datasets as follows:

- Figure 5 shows the performance of Random search, BO, and LBO (using both estimated and True $L$) for the TS acquisition function.
- Figure 6 shows the performance of Random search, BO, and LBO (using both estimated and True $L$) for the UCB acquisition function.
- Figure 7 shows the performance of Random search, BO, and LBO (using both estimated and True $L$) for the EI acquisition function.
- Figure 8 shows the performance of Random search, BO, and LBO (using both estimated and True $L$) for the PI acquisition function.
- Figure 9 shows the performance of BO and LBO using the estimated $L$ for the all acquisition function.
- Figure 10 shows the performance of Random search, BO, and LBO (using both estimated and True $L$) for the UCB acquisition function with very large $\beta = 10^{16}$. 
Figure 5: Comparing the performance of the conventional BO acquisition function, corresponding LBO mixed acquisition function, Lipschitz optimization and random exploration for the TS acquisition functions.
Figure 6: Comparing the performance of the conventional BO acquisition function, corresponding LBO mixed acquisition function, Lipschitz optimization and random exploration for the UCB acquisition functions.
Figure 7: Comparing the performance of the conventional BO acquisition function, corresponding LBO mixed acquisition function, Lipschitz optimization and random exploration for the EI acquisition functions.
Figure 8: Comparing the performance of the conventional BO acquisition function, corresponding LBO mixed acquisition function, Lipschitz optimization and random exploration for the PI acquisition functions.
Figure 9: Comparing the performance across the four BO and the corresponding LBO acquisition functions against Lipschitz optimization and random exploration on all the test functions (Better seen in color).
Figure 10: Comparing the performance of the conventional BO acquisition function, corresponding LBO mixed acquisition function, Lipschitz optimization and random exploration for the UCB acquisition functions when using very large $\beta = 10^{16}$