Combining Bayesian Optimization and Lipschitz Optimization

ECML 2020

Mohamed Osama Ahmed, Sharan Vaswani, Mark Schmidt
University of British Columbia

Bayesian optimization (BO)

- BO solves nonconvex optimization problems
  - hyperparameter tuning, control, …
- Build a probabilistic model for the objective
  - Usually using Gaussian Process
- Each iteration optimizes a cheap proxy function instead of the expensive f
  - Acquisition function decide where to sample next

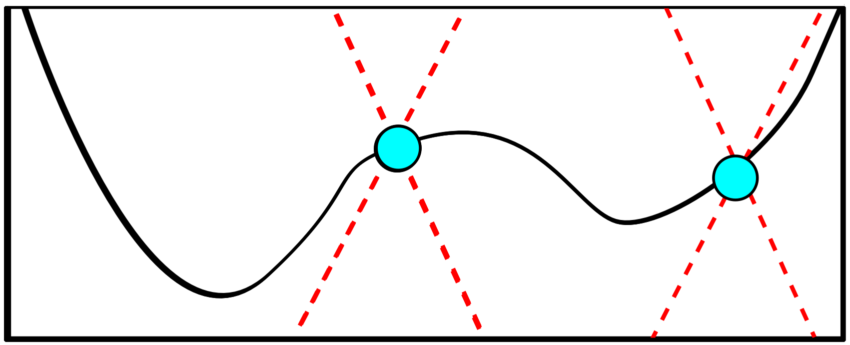
Lipschitz Continuity

- If f is L-Lipschitz continuous then

$$|f(x) - f(x_0)| \leq L||x - x_0||, \forall x, x_0 \in \mathbb{R}^d$$

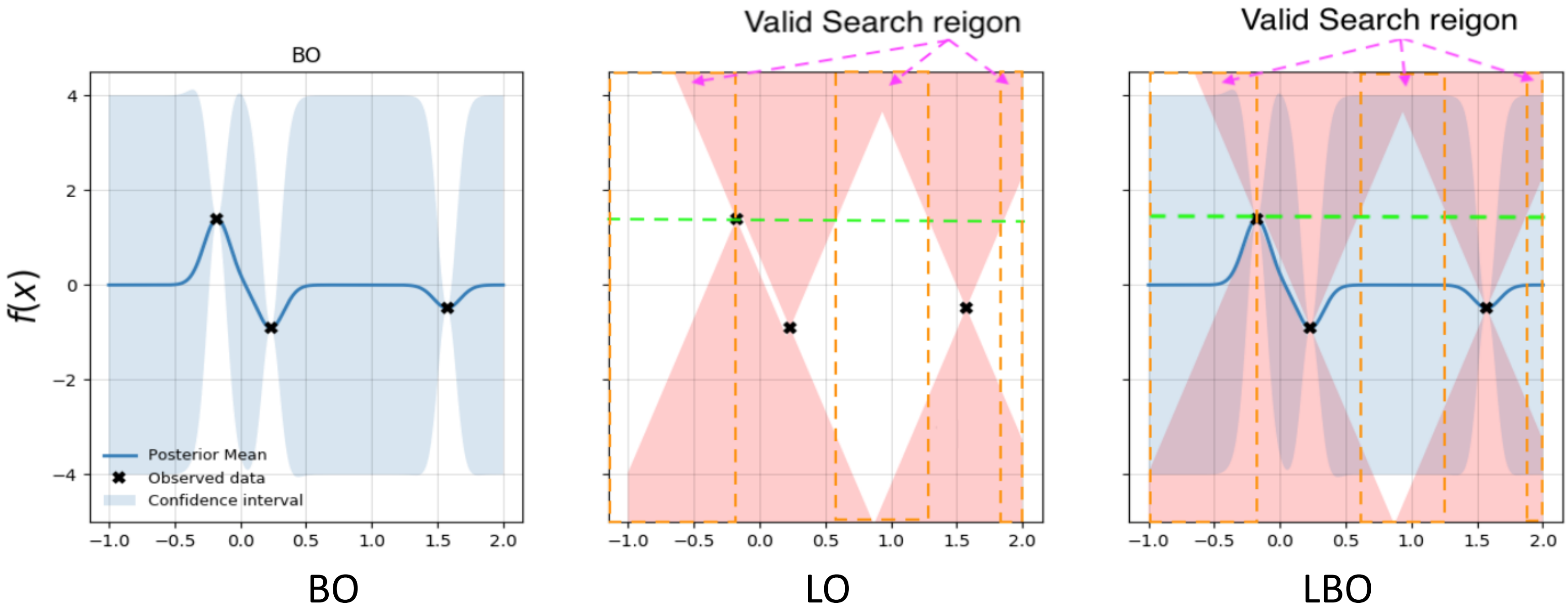
- So f can be bounded as follows:

$$f(x_0) - L\Big||x - x_0|\Big| \leq f(x) \leq f(x_0) + L||x - x_0||$$



Lipschitz BO

- How to use Lipschitz bounds to improve BO?
  - Eliminates points *x* that cannot be solutions

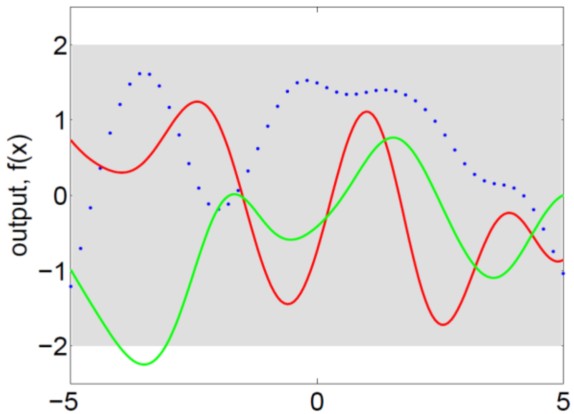


Contributions

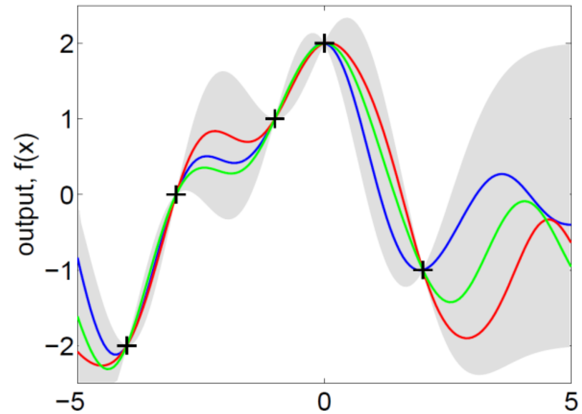
- Propose Lipschitz Bayesian optimization (LBO)
- LBO does not increase the asymptotic runtime
- Propose a simple heuristics to estimate the Lipschitz constant
  - Asymptotically, does not rule out global optimum
  - Harmless in terms of convergence speed
- Our experiments on 15 datasets with 4 acquisition functions show that LBO performs substantially better than BO
- Thompson sampling demonstrates drastic improvements
  - Lipschitz information corrected for its well-known “over-exploration” phenomenon.

BO

- Build a probabilistic model for the objective
  - Usually using Gaussian Process
  - Balances exploration and exploitation
  - Faster than random
  - Can suffer from over-exploration



Prior



Posterior

[Rasmussen 2005]

BO Algorithm

Algorithm Bayesian optimization

Input: initial vector x_0, observation y_0, $D_0 = (x_0, y_0)$.
for iteration $t = 0, 1, 2, \ldots, T$ **do**
select new x_{t+1} by optimizing acquisition function a

$$x_{t+1} = \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, a(x; D_t)$$

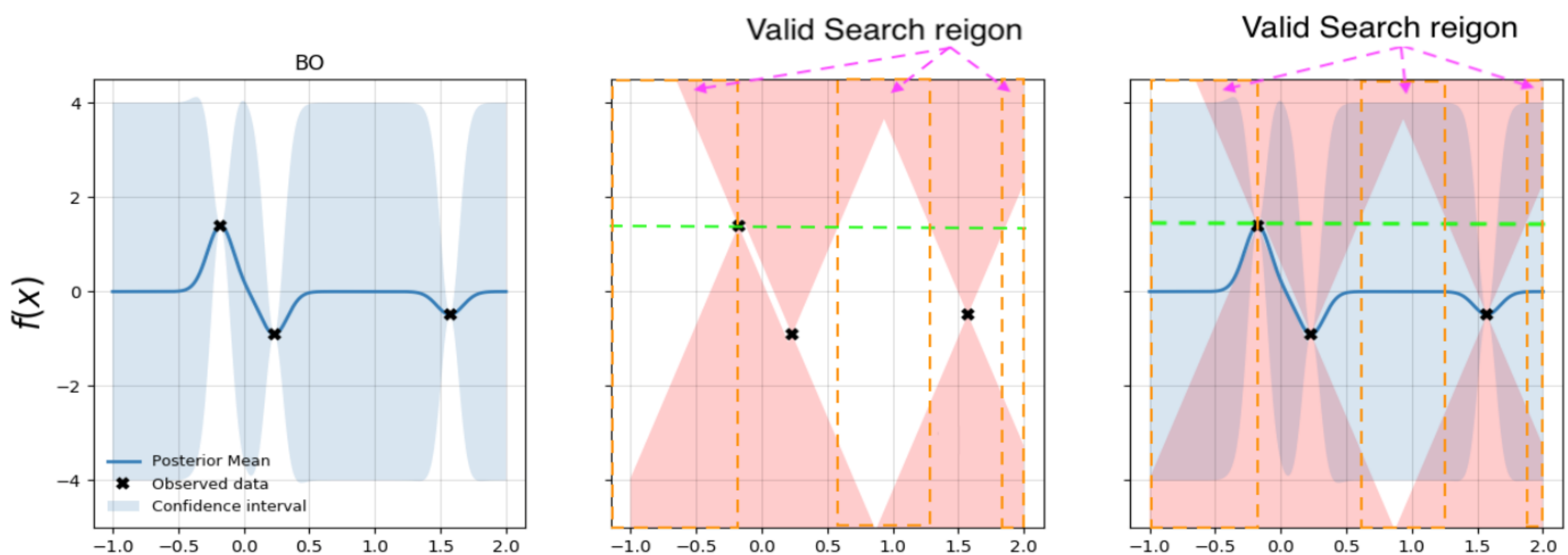
query objective function to obtain y_{n+1}.
update model by $D_{n+1} = (x_{n+1}, y_{n+1})$
end for
return location of the maximum

Lipschitz Optimization (LO)

- *L* gives a bound on the maximum amount that the function can change
- Uses the Lipschitz inequalities to prune the search space
- Faster than random
- Hard to estimate *L*

Lipschitz BO (LBO)

- How to use Lipschitz bounds to improve BO?
  - Eliminates points *x* that cannot be solutions



BO LO LBO

Harmless Lipschitz optimization

- In practice, we do not know *L.*
- Solution: calculate underestimate:

$$L_t^{lb} = \max_{i,j \in [t]; x_i \neq x_j} \left\{ \frac{|f(x_i) - f(x_j)|}{||x_i - x_j||_2} \right\}$$

- This estimate monotonically increases
  - But may rule out the solution

- Proposed solution

$$L_t^{ub} = c\, t L_t^{lb}$$

- Paper shows that such strategies are harmless.
  - Guaranteed to be at least as fast as random.

LBO strategies

- We use Lipschitz bounds in BO by modifying popular acquisition functions
  - Truncated-PI, Truncated-EI, Truncated-UCB

- We define the Lipschitz bounds as:

$$f^l = \max_i \{f(x_i) - L\|x - xi\|_2\}$$
$$f^u = \min_i \{f(x_i) + L\|x - xi\|_2\}$$

- Instead of the limits on y ϵ (-∞, ∞), we set the limits to be (L_f,U_f).
- L_f is given by:

$$L_f = \begin{cases} y^*, & \text{if } y^* \in \left(f^l(x), f^u(x)\right) \\ f^u(x), & \text{if } y^* \in \left(f^u(x), \infty\right) \end{cases}$$

- $U_f = f^u$

Experimental Setup

- GP with Matern kernel
- We use standard tricks such as standardize the function values
- Algorithms compared:
  - EI, PI, UCB and TS
  - LBO
    - TEI and TPI (Truncated EI and PI)
    - AR-UCB and AR-TS (Accept-Reject UCB and TS)
- Benchmark on standard datasets:
  - Branin, Camel, Goldstein Price, Hartmann (2 variants), Michalwicz (3 variants) and Rosenbrock (4 variants)
  - robot-pushing simulation

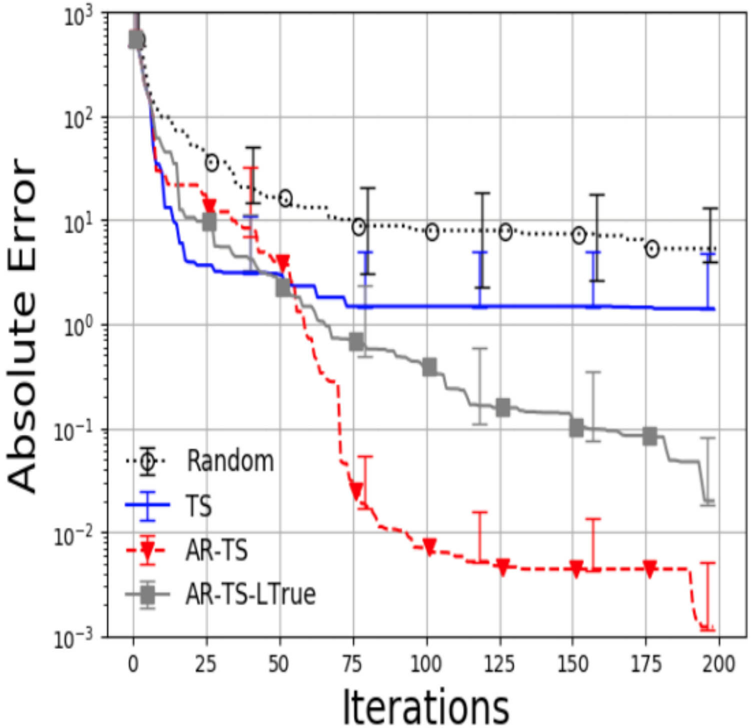
Results

- Results are divided into 4 groups:

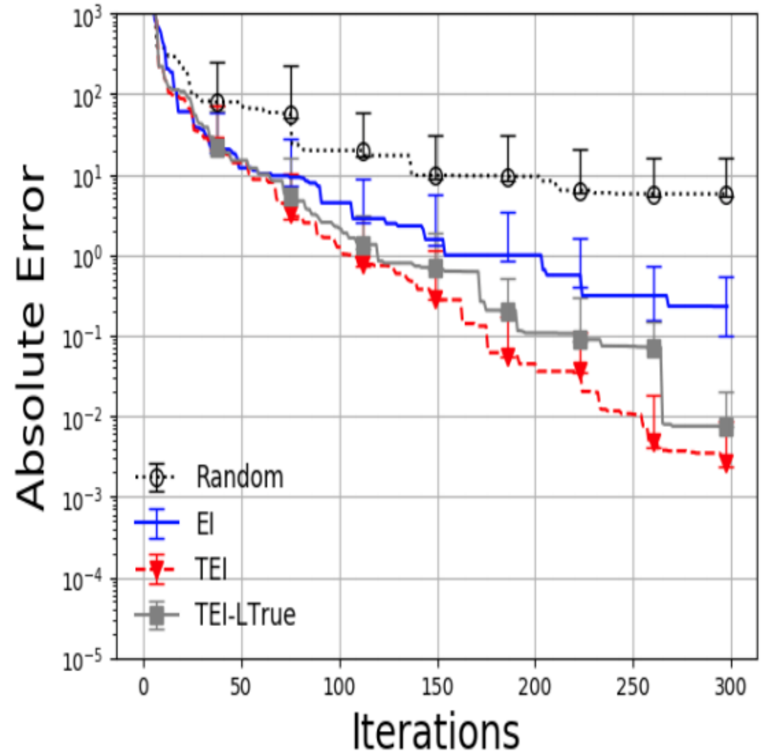
Scenario	Percentage of cases
LBO provides huge improvements over BO	21%
LBO provides improvements over BO	9%
LBO performs similar to BO	60%
LBO performs slightly worse than BO	10%

Results – Examples of Huge Gain

Rosenbrock 3D

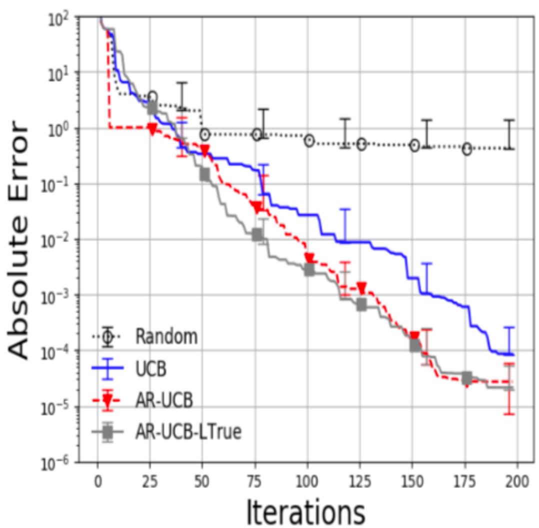


Goldstein 2D



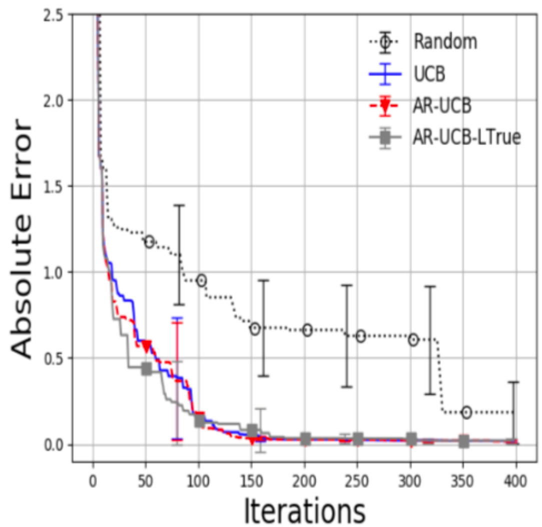
Results – Examples of Other Scenarios

Some improvement



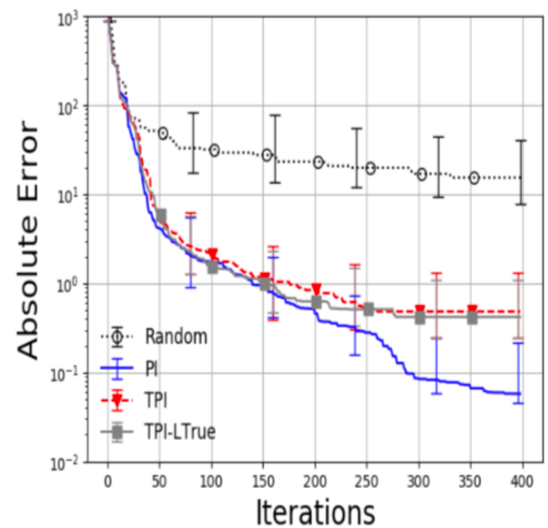
(a) Rosenbrock 2D-UCB

Same performance



(b) Robot pushing 4D-UCB

BO better



(c) Rosenbrock 4D-PI

Conclusion

- We proposed simple ways to combine Lipschitz inequalities with some of the most common BO methods.
  - "Harmless method" to overestimate Lipschitz constant.

- Experiments show that this often gives a performance gain.
  - In the worst case it performs similar to a standard BO method.