Bayesian Optimization by Density Ratio Estimation

Louis C. Tiao[1,3] **Aaron Klein**[2] **Cédric Archambeau**[2]
Edwin V. Bonilla[3,1] **Matthias Seeger**[2] **Fabio Ramos**[1,4]
[1]University of Sydney [2]Amazon Web Services [3]CSIRO's Data61 [4]NVIDIA

Abstract

Bayesian optimization (BO) is among the most effective and widely-used blackbox optimization methods. BO proposes solutions according to an explore-exploit trade-off criterion encoded in an acquisition function, many of which are derived from the posterior predictive of a probabilistic surrogate model. Prevalent among these is the expected improvement (EI). Naturally, the need to ensure analytical tractability in the model poses limitations that can ultimately hinder the efficiency and applicability of BO. In this paper, we cast the computation of EI as a binary classification problem, building on the well-known link between class-probability estimation (CPE) and density ratio estimation (DRE), and the lesser-known link between density ratios and EI. By circumventing the tractability constraints imposed on the model, this reformulation provides numerous natural advantages, not least in scalability, increased flexibility, and greater representational capacity.

1 Introduction

Bayesian optimization (BO) is a sample-efficient methodology for the optimization of expensive black-box functions [4, 23]. In brief, BO proposes candidate solutions according to an *acquisition function* that encodes the explore-exploit trade-off. At the core of BO is a probabilistic surrogate model based on which the acquisition function can be computed. The probabilistic model of choice in BO is commonly the Gaussian process (GP), owing to its flexibility and ability to yield uncertainty estimates.

However, GP-based BO can also be hampered by the limitations of GPs. Notably, they a) scale cubically with the number of observations [33] and b) assume stationarity, i.e. that the covariances between outputs are translation-invariant with respect to their inputs [26]. Further, they are not inherently equipped to deal with c) discrete variables, ordered or otherwise (i.e. categorical), and d) variables with conditional dependency structures [13]. Naturally, to address these issues, much of the focus has been directed toward extending the surrogate model itself. This has often led to ad hoc extensions that, by necessity of ensuring analytical tractability, place strong and oversimplifying assumptions at the expense of expressiveness.

Recognizing that the surrogate model is only a means to an end (i.e. of constructing an acquisition function), we seek to express acquisition functions in an alternate form that does not impose analytical tractability constraints on the surrogate model. Of particular interest is the expected improvement (EI) function [20], which has seen widespread adoption. Remarkably, Bergstra et al. [2] demonstrate that the EI function can be expressed as the *relative* ratio between two densities [34]. To estimate this density ratio, they propose a method, known as the tree-structured Parzen estimator (TPE), which can naturally deal with tree-structured inputs, discrete inputs, and scales linearly with the number of observations.

In this paper, we underscore the potential shortcomings of the TPE approach for tackling the general density ratio estimation (DRE) problem. In § 2, we highlight, among other issues that may lead to numerical instability, its tendency to scale poorly to higher dimensions [28]. In § 3, we explore

4th Workshop on Meta-Learning at NeurIPS2020, Vancouver, Canada.

more powerful alternatives to fully exploit the link between DRE and the EI function, namely, DRE by class-probability estimation (CPE). This approach retains the strengths of TPE while scaling better with the dimensionality, and enables one to build arbitrarily expressive classifiers. Depending on the choice of classifier, it is possible to capture not only non-linear, but also non-stationary and heteroscedastic behaviours. Our experiments in § 4 demonstrate that our BO-by-DRE approach, termed BORE, competes favorably with state-of-the-art blackbox optimization algorithms on a variety of challenging synthetic test problems and meta-surrogate benchmarks for automated machine learning (AUTOML) [16].

2 Background

Let $\mathbf{x} \in \mathcal{X}$ denote an input to the blackbox function $f : \mathcal{X} \to \mathbb{R}$ and $y \sim \mathcal{N}(f(\mathbf{x}), \sigma^2)$ the noisy observation of the corresponding output with noise variance σ^2. Further, let the observations be denoted by $\mathcal{D}_N = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$.

Expected improvement. First we define a threshold $\tau = \Phi^{-1}(\gamma)$ where constant γ denotes some quantile of the observed y values, i.e. $\gamma = \Phi(\tau) = p(y < \tau \,|\, \mathcal{D}_N)$. Let $I_\gamma(\mathbf{x})$ be the utility function that quantifies the non-negative improvement over τ

$$I_\gamma(\mathbf{x}) = \max(\tau - y, 0). \tag{1}$$

Then, the EI function $\alpha_\gamma(\mathbf{x}; \mathcal{D}_N)$ is defined as the expected value of $I_\gamma(\mathbf{x})$ under the posterior predictive distribution $p(y \,|\, \mathbf{x}, \mathcal{D}_N)$

$$\alpha_\gamma(\mathbf{x}; \mathcal{D}_N) = \mathbb{E}_{p(y \,|\, \mathbf{x}, \mathcal{D}_N)}[I_\gamma(\mathbf{x})]. \tag{2}$$

Note that the dependence of $\alpha_\gamma(\mathbf{x}; \mathcal{D}_N)$ on γ occurs in τ (which is implicitly a function of γ). For example, $\gamma = 0$ leads to the conventional setting of $\tau = \min_n y_n$. In what follows, we will consider a relaxation of τ with settings where $\gamma > 0$.

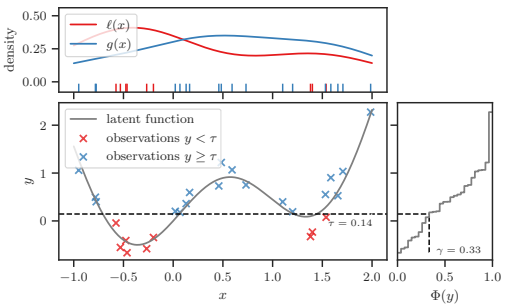


Figure 1: Optimizing a synthetic test function $f(x) = \sin(3x) + x^2 - 0.7x$ with observation noise $\varepsilon \sim \mathcal{N}(0, 0.2^2)$. Initially, $N = 27$ candidates values of x are drawn from $[-2, 1]$. The candidates whose corresponding target y values are in the top performing fraction $\gamma = 1/3$ are shown in *blue*. The remaining candidates are shown in *orange*. The density estimates of $\ell(\mathbf{x})$ and $g(\mathbf{x})$ are shown in the top pane.

Relative density ratio. Let $\ell(\mathbf{x})$ and $g(\mathbf{x})$ be a pair of densities. The *γ-relative density ratio* of $\ell(\mathbf{x})$ and $g(\mathbf{x})$ is defined as the ratio of $\ell(\mathbf{x})$ and the *γ-mixture density* $\gamma\ell(\mathbf{x}) + (1 - \gamma)g(\mathbf{x})$,

$$r_\gamma(\mathbf{x}) = \frac{\ell(\mathbf{x})}{\gamma\ell(\mathbf{x}) + (1 - \gamma)g(\mathbf{x})}, \tag{3}$$

for some mixing proportion $0 \leq \gamma < 1$ [34]. For $\gamma = 0$, we recover the ordinary density ratio $r_0(\mathbf{x}) = \ell(\mathbf{x})/g(\mathbf{x})$. Before moving on, observe that we can directly express $r_\gamma(\mathbf{x})$ as a function of $r_0(\mathbf{x})$,

$$r_\gamma(\mathbf{x}) = (\gamma + r_0(\mathbf{x})^{-1}(1 - \gamma))^{-1}. \tag{4}$$

Now, suppose $\ell(\mathbf{x})$ and $g(\mathbf{x})$ are distributions such that $\mathbf{x} \sim \ell(\mathbf{x})$ if $y < \tau$, and $\mathbf{x} \sim g(\mathbf{x})$ if $y \geq \tau$. Then, let us express the conditional $p(\mathbf{x} \,|\, y, \mathcal{D}_N)$ in terms of $\ell(\mathbf{x})$ and $g(\mathbf{x})$

$$p(\mathbf{x} \,|\, y, \mathcal{D}_N) = \begin{cases} \ell(\mathbf{x}) & \text{if } y < \tau, \\ g(\mathbf{x}) & \text{if } y \geq \tau. \end{cases} \tag{5}$$

Remarkably, under this construction, as illustrated in Figure 1, Bergstra et al. [2] demonstrate that the EI function can be expressed as the γ-relative density ratio, up to some constant multiplicative factor

$$\boxed{\alpha_\gamma(\mathbf{x}; \mathcal{D}_N) \propto r_\gamma(\mathbf{x})} \tag{6}$$

For completeness, we provide the self-contained derivation in **??**. Hence, this result shows that the problem of maximizing EI reduces to that of maximizing the γ-relative density ratio,

$$\mathbf{x}^\star = \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha_\gamma(\mathbf{x}; \mathcal{D}_N) = \arg\max_{\mathbf{x} \in \mathcal{X}} r_\gamma(\mathbf{x}), \tag{7}$$

for which a wide variety of approaches are available [29].

2.1 Tree-structured Parzen estimator

In practice, to solve the optimization problem of eq. 7, Bergstra et al. [2] propose taking the following approach:

1. Since the relative density ratio $r_\gamma(\mathbf{x})$ is a monotonically non-decreasing function of the ordinary density ratio $r_0(\mathbf{x})$, they restrict their attention to maximizing the latter,

$$\mathbf{x}^\star = \arg\max_{\mathbf{x}\in\mathcal{X}} r_0(\mathbf{x}), \tag{8}$$

   thus, effectively ignoring the mixing proportion γ altogether.

2. Then, they estimate the ordinary density ratio $r_0(\mathbf{x})$ by separately estimating its constituent numerator $\ell(\mathbf{x})$ and denominator $g(\mathbf{x})$ using a tree-based variant of kernel density estimation (KDE) [25].

It is easy to see why this approach might be favorable compared to methods based on GP regression: one now incurs an $\mathcal{O}(N)$ computational cost as opposed to the $\mathcal{O}(N^3)$ cost of GP posterior inference. Furthermore, it is equipped to deal with tree-structured, mixed continuous, ordered and unordered discrete inputs. In spite of its advantages, TPE is not without potential pitfalls as discussed next.

2.2 Potential pitfalls

The first major drawback of TPE lies within step 1:

Singularities. Relying on the ordinary density ratio can result in numerical instabilities since it is unbounded, and often diverges to infinity even in simple toy scenarios. In contrast, the γ-relative density ratio is always bounded above by γ^{-1} when $\gamma > 0$ [34].

The other potential problems of the TPE lie within step 2:

Vapnik's principle. Conceptually, independently estimating the densities is actually a more cumbersome approach that violates Vapnik's principle—namely, that when solving a problem of interest, one should refrain from resorting to solve a more general problem as an intermediate step [32]. In this instance, *density* estimation is a more general problem that is arguably more difficult than *density ratio* estimation.

Kernel bandwidth. KDE depends crucially on the selection of an appropriate kernel bandwidth, which is notoriously difficult [21, 24]. Furthermore, even with an optimal selection of a single fixed bandwidth, it cannot cannot simultaneously adapt to low- and high-density regions [31].

Error sensitivity. These difficulties are exacerbated by the fact that one is required to select *two* bandwidths, whereby the optimal bandwidth for an individual density may not necessarily be appropriate for approximating the density ratio. Indeed, it may even have deleterious effects. This makes this approach unforgiving to any error in the individual approximations, particularly in that of the denominator $g(\mathbf{x})$, which has an outsized influence on the resulting density ratio.

Curse of dimensionality. For these reasons and more, KDE often falls short in high-dimensional regimes. In contrast, direct DRE has consistently proven to scale better with dimensionality [28].

Optimization. Ultimately, we care not only about *estimating* the density ratio, but also *maximizing* it wrt to inputs for the purposes of candidate suggestion. It is cumbersome to maximize the ratio of KDEs and one must typically resort to derivative-free methods.

3 Methodology

We consider an alternative approach to tackling the optimization problem of eq. 7 that circumvents the issues of TPE outlined in § 2.2, namely one based on CPE. Density ratio estimation is closely-linked to CPE [3, 6, 19, 22, 29]. To see this, let us introduce binary target variables $z = \mathbb{I}[y < \tau]$ or, more explicitly,

$$z = \begin{cases} 1 & \text{if } y < \tau, \\ 0 & \text{if } y \geq \tau. \end{cases} \tag{9}$$

Algorithm 1: Bayesian optimization by density ratio estimation (BORE).

1 **while** *under budget* **do**
2 $\quad \boldsymbol{\theta}^* \leftarrow \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ // update classifier by optimizing parameters $\boldsymbol{\theta}$ wrt BCE loss
3 $\quad \mathbf{x}_N \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}} \pi_{\boldsymbol{\theta}^*}(\mathbf{x})$ // suggest new candidate by optimizing input $\mathbf{x}$ wrt classifier output
4 $\quad y_N \leftarrow f(\mathbf{x}_N)$ // obtain y_N by evaluating blackbox function at $\mathbf{x}_N$
5 $\quad \mathcal{D}_N \leftarrow \mathcal{D}_{N-1} \cup \{(\mathbf{x}_N, y_N)\}$ // update dataset
6 **end**

By definition, we have $\ell(\mathbf{x}) = p(\mathbf{x} \,|\, z = 1)$ and $g(\mathbf{x}) = p(\mathbf{x} \,|\, z = 0)$. Then, we can apply Bayes' rule to give

$$r_0(\mathbf{x}) = \frac{\ell(\mathbf{x})}{g(\mathbf{x})} = \frac{p(\mathbf{x} \,|\, z = 1)}{p(\mathbf{x} \,|\, z = 0)} = \frac{p(z = 0)}{p(z = 1)} \frac{p(z = 1 \,|\, \mathbf{x})}{p(z = 0 \,|\, \mathbf{x})}. \tag{10}$$

Now, by construction, we have

$$\frac{p(z = 0)}{p(z = 1)} = \left(\frac{\gamma}{1 - \gamma}\right)^{-1} \qquad \text{and} \qquad \frac{p(z = 1 \,|\, \mathbf{x})}{p(z = 0 \,|\, \mathbf{x})} = \frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})}, \tag{11}$$

where $\pi(\mathbf{x}) = p(z = 1 \,|\, \mathbf{x})$ denotes the class-posterior probability. Hence,

$$r_0(\mathbf{x}) = \left(\frac{\gamma}{1 - \gamma}\right)^{-1} \frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})}. \tag{12}$$

We plug this into eq. 4 to give

$$\boxed{r_\gamma(\mathbf{x}) = \gamma^{-1} \pi(\mathbf{x})} \tag{13}$$

We refer to **??** for derivations. Equations (12) and (13) establish the precise link between the class-posterior probability and the ordinary and γ-relative density ratios, respectively. Notice in particular that the γ-relative density ratio is exactly equivalent to the class-posterior probability up to constant factor γ^{-1}.

Let us estimate the class-posterior probability $\pi(\mathbf{x})$ using $\pi_{\boldsymbol{\theta}}(\mathbf{x})$, a function parameterized by $\boldsymbol{\theta}$. Then, we can approximate the γ-relative density ratio with $r_\gamma(\mathbf{x}) \simeq \gamma^{-1} \pi_{\boldsymbol{\theta}}(\mathbf{x})$. To recover the true class-posterior probability, one can minimize a *proper scoring rule* [9] such as the binary cross-entropy (BCE) loss:

$$\mathcal{L}(\boldsymbol{\theta}) = -\beta \cdot \mathbb{E}_{\ell(\mathbf{x})}[\log \pi_{\boldsymbol{\theta}}(\mathbf{x})] - (1 - \beta) \cdot \mathbb{E}_{g(\mathbf{x})}[\log (1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}))], \tag{14}$$

where β denotes the class balance rate. It can be verified that the BCE loss attains its minimum at $\boldsymbol{\theta}^*$ such that

$$\pi_{\boldsymbol{\theta}^*}(\mathbf{x}) = \frac{\beta \ell(\mathbf{x})}{\beta \ell(\mathbf{x}) + (1 - \beta) g(\mathbf{x})}. \tag{15}$$

We refer to **??** for detailed derivations. Now, since $\beta = \gamma$ by construction, this leads to $\pi_{\boldsymbol{\theta}^*}(\mathbf{x}) = \gamma \cdot r_\gamma(\mathbf{x})$. We provide an illustration on a toy example in **??**.

Hence, in the so-called BO loop (summarized in Algorithm 1), we alternately optimize the classifier parameters $\boldsymbol{\theta}$ wrt to the BCE loss (to improve its approximation to the true class-posterior probability; Line 2) and the classifier input $\mathbf{x}$ wrt to its output (to suggest the next candidate to evaluate; Line 3).

In traditional GP-based methods, Line 3 typically consists of maximizing the EI function, explicitly expressed as a combination of the properties of the GP posterior predictive (its mean, variance, pdf and cdf), while Line 2 would be the optimization of the GP hyperparameters wrt the marginal likelihood. By analogy with our approach, the parameterized function $\pi_{\boldsymbol{\theta}}(\mathbf{x})$ is *itself* an approximation to the EI function to be maximized directly, while the approximation is tightened by improving its fit to the true class-posterior probabilities, in turn through optimization wrt the BCE loss.

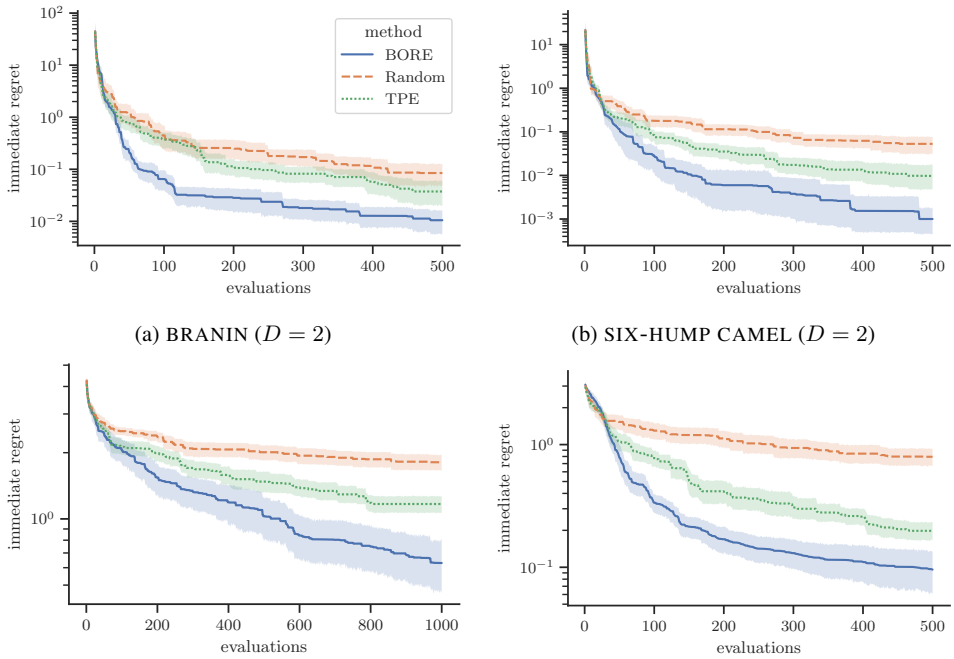
In short, we have reduced the problem of computing EI to that of training a probabilistic classifier, thus unlocking a broad range of possible alternatives to GPs. This enables one to employ virtually any state-of-the-art classification method available to parameterize the classifier with arbitrarily expressive approximators that have the capacity to deal with non-linear, non-stationary, and heteroscedastic phenomena commonly encountered in BO.

In this work, we parameterize $\pi_{\boldsymbol{\theta}}(\mathbf{x})$ by a feed-forward neural network (NN). This is an attractive choice not only for its universal approximation guarantees [11] but because a) one can easily adopt stochastic gradient descent (SGD) methods to scale up its parameter learning [17], and b) it is differentiable end-to-end, which enables the use of quasi-Newton methods such as L-BFGS [18] for candidate suggestion.

4 Experiments

We describe the experiments conducted to empirically evaluate our method. The classifier $\pi_{\boldsymbol{\theta}}(\mathbf{x})$ is a multi-layer perceptron (MLP), with 2 hidden layers, each with 32 units. We consistently found `elu` activations [7] to be particularly effective for low-dimensional problems, with `relu` remaining otherwise the best choice. We optimize the weights with ADAM [14] using batch size of $B = 64$. For candidate suggestion, we optimize the input of the classifier wrt to its output using multi-started L-BFGS with three random restarts. To encourage exploration, we suggest random candidates at a proportion $\epsilon = 0.1$ of the time. Further details concerning the implementation and setting of additional hyperparameters are provided in **??**.

Synthetic test functions. We first consider a number of challenging synthetic test functions for optimization [30], namely, the BRANIN, SIX-HUMP CAMEL, MICHALEWICZ5D, and HARTMANN6D functions. To quantitatively assess performance, we report the *immediate regret*, defined as the absolute error between the global minimum and the lowest function value attained thus far. To assess the sample efficiency, we compare the immediate regret over the number of function evaluations against baselines. The baselines shown here are Random Search and TPE, as implemented in the `HyperOpt` library [1]. For each method, we show the mean and 95% confidence interval (CI) across results obtained from 20 repeated runs.



(a) BRANIN ($D = 2$)

(b) SIX-HUMP CAMEL ($D = 2$)

(c) MICHALEWICZ5D ($D = 5$)

(d) HARTMANN6D ($D = 6$)

Figure 2: Immediate regret over function evaluations on the synthetic test problems.

The results are shown in Figure 2. Across these problems, we find BORE to be competitive against TPE. In the BO folklore, TPE is known to have a tendency of over-exploitation. This is especially manifest in the SIX-HUMP CAMEL problem, which is designed to have multiple local minima. In visualizations included in **??**, we show that TPE consistently gets bogged down in a local minimum,

while BORE is able to balance the exploitation of the various local minima with the exploration of other parts of the space.

Meta-surrogate benchmarks. Finally, we compare against a range of state-of-the-art optimization methods on the meta-surrogate benchmarks, Profet [16]. First, BO methods with differing probabilistic models: GP-based BO (GP-BO) [15], the random forest-based SMAC [12] and TPE [2], and second, two evolutionary algorithms: CMA-ES [10] and differential evolution (DE) [27]. Profet emulates the hyperparameter tuning of common machine learning algorithms such as support vector machines (SVMs) [8], gradient boosting (XGBoost) [5], and others, on classification and regression problems, by sampling tasks (in the form of objective functions) from a generative meta-model. We sampled 50 tasks for each of the three benchmark classes: META-SVM, META-FCNET, and META-XGBoost. Each optimizer was evaluated on each task with 20 independent runs using different random seeds. To aggregate the performance across tasks, we follow the protocol of Klein et al. [16] and report the average ranks and the empirical cdf (ECDF) of the runtime, using a single run of Random Search with 200 iterations as targets, cf. [16] for further details.

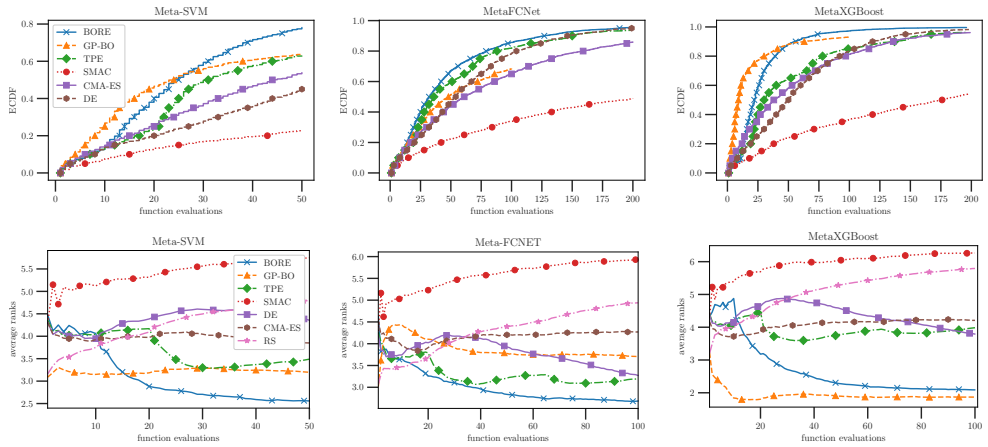


Figure 3: ECDFs (top) and ranks (bottom) of the three problem types (SVM, FCNET, and XGBoost) of Profet. For each problem type, results are aggregated over 20 runs per method on 50 task sampled from the generative meta-model.

Figure 3 shows the results on all benchmark classes, from which we see that BORE consistently performs better than all other baselines. In particular, observe that despite GP-BO approaching the optimum faster in the early stages, it is eventually outperformed by BORE after having observed a sufficient amount of data. Lastly, note that we were only able to run GP-BO for 100 function evaluations on the META-FCNET and META-XGBoost benchmarks due to its prohibitively-high computational overhead.

5 Conclusion

We have presented a novel methodology for BO based on the observation that the problem of computing EI can be reduced to that of probabilistic classification. This observation is made through the well-known link between CPE and DRE, and the lesser-known insight that EI can be expressed as a relative density ratio between two unknown distributions. We discussed important ways in which TPE, an early attempt to exploit the latter, falls short. Further, we demonstrated that a prototype implementation of this methodology, based on a simple feed-forward NN, can outperform TPE and be competitive with state-of-the-art derivative-free optimization methods.

A key appeal of this methodology lies in the room it allows for variations. Indeed, any other state-of-the-art classification method can readily be applied. In particular, SVMs, random forests, and XGBoost may prove to be strong contenders against NNs. Another axis of variation worth exploring is the potential benefits that other direct DRE methods may have to offer. In future work, we will also explore the effects of a fully-Bayesian treatment of the classifier parameters.

Overall, the simplicity and effectiveness of BORE make it a promising BO approach, and its extensibility offers numerous future directions.

Acknowledgments and Disclosure of Funding

L. Tiao is supported by an Australian Government Research Training Program (RTP) Scholarship and a CSIRO Data61 Postgraduate Scholarship.

References

[1] J. Bergstra, D. Yamins, and D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123, 2013.

[2] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

[3] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88, 2007.

[4] E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[5] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[6] K. F. Cheng, C.-K. Chu, et al. Semiparametric density estimation under a two-sample density ratio model. *Bernoulli*, 10(4):583–604, 2004.

[7] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[10] N. Hansen. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*. Springer Berlin Heidelberg, 2006.

[11] K. Hornik, M. Stinchcombe, H. White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[12] F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the Fifth International Conference on Learning and Intelligent Optimization (LION'11)*, 2011.

[13] R. Jenatton, C. Archambeau, J. González, and M. Seeger. Bayesian optimization with tree-structured dependencies. In *International Conference on Machine Learning*, pages 1655–1664, 2017.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] A. Klein, S. Falkner, N. Mansur, and F. Hutter. Robo: A flexible and robust bayesian optimization framework in python. In *NIPS Workshop on Bayesian Optimization (BayesOpt'17)*, 2017.

[16] A. Klein, Z. Dai, F. Hutter, N. Lawrence, and J. Gonzalez. Meta-surrogate benchmarking for hyperparameter optimization. In *Proceedings of the 32th International Conference on Advances in Neural Information Processing Systems (NIPS'19)*, 2019.

[17] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[18] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[19] A. Menon and C. S. Ong. Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*, pages 304–313, 2016.

[20] J. Mockus, V. Tiesis, and A. Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2, 1978.

[21] B. U. Park and J. S. Marron. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, 85(409):66–72, 1990.

[22] J. Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85 (3):619–630, 1998.

[23] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[24] S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(3):683–690, 1991.

[25] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[26] J. Snoek, K. Swersky, R. Zemel, and R. Adams. Input warping for bayesian optimization of non-stationary functions. In *International Conference on Machine Learning*, pages 1674–1682, 2014.

[27] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997.

[28] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440, 2008.

[29] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[30] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved August 27, 2020, from `http://www.sfu.ca/~ssurjano`, 2013.

[31] G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

[32] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[33] C. K. Williams and C. E. Rasmussen. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520, 1996.

[34] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama. Relative density-ratio estimation for robust distribution comparison. In *Advances in neural information processing systems*, pages 594–602, 2011.