CORRECTING BOUNDARY OVER-EXPLORATION DEFICIENCIES IN BAYESIAN OPTIMIZATION WITH VIRTUAL DERIVATIVE SIGN OBSERVATIONS

Eero Siivola[1]*, Aki Vehtari*[1]*, Jarno Vanhatalo*[2]*, Javier González*[3]*, Michael Riis Andersen*[1]

[1]Aalto University, Dept. of Computer Science, [2]University of Helsinki,
Dept. of Math. and Stat., and Dept. of Biosciences, [3]Amazon.com

ABSTRACT

Bayesian optimization (BO) is a global optimization strategy designed to find the minimum of an expensive black-box function, typically defined on a compact subset of $\mathcal{R}^d$, by using a Gaussian process (GP) as a surrogate model for the objective. Although currently available acquisition functions address this goal with different degree of success, an over-exploration effect of the contour of the search space is typically observed. However, in problems like the configuration of machine learning algorithms, the function domain is conservatively large and with a high probability the global minimum does not sit on the boundary of the domain. We propose a method to incorporate this knowledge into the search process by adding virtual derivative observations in the GP at the boundary of the search space. We use the properties of GPs to impose conditions on the partial derivatives of the objective. The method is applicable with any acquisition function, it is easy to use and consistently reduces the number of evaluations required to optimize the objective irrespective of the acquisition used. We illustrate the benefits of our approach in an extensive experimental comparison.

***Index Terms*—** Bayesian optimization, Gaussian process, virtual derivative sign observation.

1. INTRODUCTION

Global optimization is a common problem in a very broad range of applications. Formally, it is defined as finding $\mathbf{x}_{\text{min}} \in \mathcal{X} \subset \mathcal{R}^d$ such that

$$\mathbf{x}_{\text{min}} = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}), \tag{1}$$

where $\mathcal{X}$ is generally considered to be a compact set of a Euclidean space. In this work, we focus on cases in which f is a black-box function whose explicit form is unknown and that it is expensive to evaluate. Thus, the goal is to locate $\mathbf{x}_{\text{min}}$ within a finite and typically small number of evaluations, which transform the original *optimization* problem in a sequence of *decision* problems.

Bayesian optimization of black-box functions using Gaussian Processes (GPs) as surrogate priors has become popular in recent years (see, e.g. [1]). Treating the decision of where to evaluate the function f next as a statistical *inference* problem has been proven effective. This is typically using an acquisition function that balances *exploration* and *exploitation*.

A common problem that has not been systematically studied in the BO literature is the tendency of most acquisition strategies to over-explore the boundary of the function domain $\mathcal{X}$. This issue is not relevant if the global minimum may lie on the border of the search space but in most cases, including when the search space is unbounded, this is not the case [2]. This effect has also been observed in the active learning literature and it is known to appear when the search is done myopically, as it is the case in most acquisitions functions [3]. Non-myopic approaches in BO can potentially deal with this problem but they are typically very computationally expensive [4].

In this paper we propose a new approach to correct the *boundary over-exploration effect* of most acquisitions without increasing the computational overhead of currently available non-myopic methods. We demonstrate that when the local minimum is known not to lie on the boundary of $\mathcal{X}$, this information can be embedded into the model of the objective function. The assumption that x_{min} does not sit on the boundary of the domain implies that the gradient of the underlying function points away from the centre on the boundary. This property of the function can be incorporated into the model using *virtual derivative observations*. Virtual derivative observations are unobserved data about the partial derivatives of the function that are treated similarly as true observations. In other words, we add pseudo derivative observations to the training set to induce the desired behaviour of the function at the boundary. As the derivative of a GP is also a GP, including virtual derivative observations in GPs is feasible with standard inference methods [5]. In this work, we demonstrate that this reduces the number of required function evaluations giving rise to a battery of more efficient BO methods. The concept of augmenting data with virtual derivative observations is illustrated in Figure 1.

The unwanted boundary over-exploration effect of the reg-

Correspondence to `eero.siivola@aalto.fi`
Work done while Javier González was at the University of Sheffield

978-1-5386-5477-4/18/$31.00 ©2018 IEEE

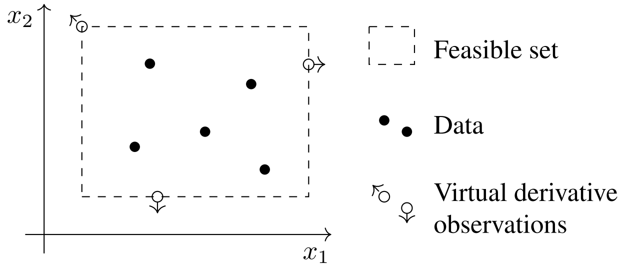


Fig. 1: The concept of virtual derivative observations in two dimensional space. The arrows indicate the direction of the observed gradient. As there are no local minima on the borders, the gradient of the function always has a non-zero component pointing outwards from the feasible set. On the corners, both partial derivatives are non-zero.

ular BO is illustrated in Figure 2 (a). A simple function consisting of two Gaussian components is optimized with the standard BO and the proposal of this work. The correction of the over-exploration effect is evident.

1.1. Related Work

Derivative observations have been used before in the BO and GP context to find minimum energy path transitions of atomic rearrangements and to decrease the number of observations needed for finding the function optimum [6, 7].

They can also be used to provide shape priors. To constrain a function to have a mode in a specified location, virtual observations of first derivative being zero and the second derivative being negative can be used [8]. Virtual derivative observations where only the sign of the derivative is known can be used to add monotonicity information [9]. To handle inference for the non-Gaussian contribution of the derivative sign information, rejection sampling, expectation propagation (EP), and Markov chain Monte Carlo (MCMC) have been used [8, 9, 10].

Surprisingly, over-exploration of boundaries has not been systematically studied before. A naive approach is to use a quadratic mean function to penalize the search in the boundary. However, this has strong limitations when the optimized function is multimodal or far away from being quadratic [9].

1.2. Contributions

The main contributions of this work are:

- A new approach for Bayesian optimization that corrects the over-exploration of the boundary of most acquisitions. The method is simple to use, can be combined with any acquisitions and always work equally or better than the standard approach. After a review of the needed background, the method is described in Section 3.
- A publicly available code framework[1] that contains an efficient implementation of the methods described in this work.
- A comprehensive analysis of the performance of the proposed method in a variety of scenarios that should give the reader a precise idea about the (i) the loss in efficiency incurred in standard methods due to the boundary over-exploration and (ii) how this issue is significantly relieved with our proposal.

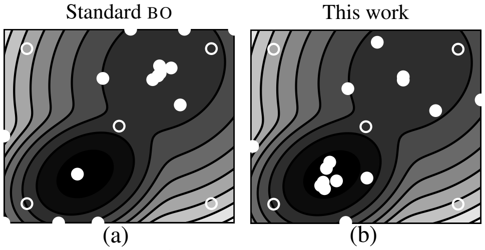


Fig. 2: Sequence of 15 evaluations when optimizing a combination of Gaussians (darker colors represent lower function values) with (a) standard BO (b) and the proposal of this work. The five white open circles are the points used to initialise the GP. The 15 white balls are the acquisitions. The GP-LCB acquisition function was used in both cases (see Section 2.2 for details). With the new proposal, fewer evaluations are spent in the boundary, and more points are collected around the global optimum.

In addition to the previous points Section 5 contains the main lessons learned in this work.

2. BACKGROUND AND PROBLEM SET-UP

The main iterative steps of any BO algorithm are: (i) Model the objective function with GP prior, which is updated based on the current set of function evaluations. (ii) Use an acquisition function, that depends on the posterior for the objective function, to decide what the next query point should be. Next, we visit both of them and detail how information from derivative observations can be naturally incorporated in the loop.

2.1. Standard GP Surrogate for Modeling of f

At iteration $n+1$, we assume that we have evaluated the objective function n times providing us the data $D = \{y^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^{n}$ where $y^{(i)}$ is, the possibly noisy, function evaluation at input location $\mathbf{x}^{(i)}$. To combine our previous knowledge about f with the dataset D, we use a GP to model f. In particular, a GP prior is directly specified on the latent function with prior assumptions encoded in the covariance function $k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$, which specifies the covariance of two latent function values $f(\mathbf{x}^{(1)})$ and $f(\mathbf{x}^{(2)})$. A zero mean Gaussian process prior

$$p(\mathbf{f}) = \mathrm{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}), \tag{2}$$

[1] Framework available at https://github.com/esiivola/vdsobo

is chosen, where $\mathbf{K}$ is a covariance matrix between n latent values $\mathbf{f}$ at input used for training, $\mathbf{X} = \big(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\big)$, s.t. $\mathbf{K}_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

In regression, n noisy observations $\mathbf{y}$ and o latent function values $\mathbf{f}_*$ at the test inputs $\mathbf{X}_*$ are assumed to have a joint Gaussian distirbution. With the noise variance σ^2, the covariance between the latent values at the training and test inputs $\mathbf{K}_*$, the covariance matrix of the latent values at the test inputs $\mathbf{K}_{**}$ and n dimensional identity matrix $\mathbf{I}$, the joint distribution of the observations and latent values at the test inputs is

$$\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_*\end{bmatrix} \sim \mathrm{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}+\sigma^2\mathbf{I} & \mathbf{K}_*^{\mathrm{T}}\\ \mathbf{K}_* & \mathbf{K}_{**}\end{bmatrix}\right). \tag{3}$$

Using the Gaussian conditioning rule, the predictive distribution for $\mathbf{f}_*$ can easily be computed and the predictive distribution of the GP can be written explicitly for any point in the domain.

2.2. Acquisition Policy

In this work, we concentrate on the *lower confidence bound* (LCB) acquisition function that minimizes the regret over the optimization area [11]. Although this is one of the most widely used acquisition function, it suffers from the over exploration effect described in the introduction of this work. As we will detail later, the ability of GPs to handle derivative observations will be key to correct this effect.

2.3. Incorporating Partial Derivative Observations in the Loop

Since the differentiation is a linear operator, the partial derivative of a (mean-square differentiable) Gaussian process remains a Gaussian process [12]. Thus, using partial derivative values for prediction and making predictions about the partial derivatives at a given point is easy to incorporate in the model and in the BO search. Since

$$\mathrm{cov}\left(\frac{\partial f^{(i)}}{\partial x_g^{(i)}},\ f^{(j)}\right) = \frac{\partial}{\partial x_g^{(i)}}\ \mathrm{cov}\Big(f^{(i)},\ f^{(j)}\Big),$$
$$\mathrm{cov}\left(\frac{\partial f^{(i)}}{\partial x_g^{(i)}},\ \frac{\partial f^{(j)}}{\partial x_h^{(j)}}\right) = \frac{\partial^2}{\partial x_g^{(i)}\partial x_h^{(j)}}\ \mathrm{cov}\Big(f^{(i)},\ f^{(j)}\Big)$$

covariance matrices in Equations (2) and (3) can be extended to include partial derivatives either as observations or as values to be predicted.

Following Riihimäki and Vehtari (2010) ([9]), denote by $m \in \{-1, 1\}$ the partial derivative value in the dimension j at $\tilde{\mathbf{x}}$. Then the probability of observing partial derivative is modelled using probit likelihood with a control parameter ν

$$p\left(m\,\middle|\,\frac{\partial \tilde{f}}{\partial \tilde{x}_j}\right) = \Phi\left(\frac{\partial \tilde{f}}{\partial \tilde{x}_j}\frac{m}{\nu}\right),\ \text{where } \Phi(z) = \int_{-\infty}^{z} N(t\,|\,0,1)\mathrm{d}t. \tag{4}$$

Let $\mathbf{m}$ be a vector of q partial derivative values at $\tilde{\mathbf{X}} = \big(\tilde{\mathbf{x}}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(q)}\big)$, $\mathbf{j}$ be a vector of the dimensions of the partial derivatives and $\tilde{\mathbf{f}}$ be the vector of latent values at $\tilde{\mathbf{X}}$ and let the partial derivatives of latent values be $\tilde{\mathbf{f}}'$. Assuming conditional independence given the latent derivative values, the likelihood becomes

$$p(\mathbf{m} \mid \tilde{\mathbf{f}}') = \prod_{i=1}^{q} \mathbf{\Phi}\left(\frac{\partial \tilde{\mathbf{f}}^{(i)}}{\partial \tilde{\mathbf{x}}_{\mathbf{j}^{(i)}}^{(i)}}\frac{m^{(i)}}{\nu}\right).$$

With function values at $\mathbf{X}$ and partial derivative values at $\tilde{\mathbf{X}}$, the joint prior for $\mathbf{f}$ and $\tilde{\mathbf{f}}$ then becomes

$$p\left(\begin{bmatrix}\mathbf{f}\\ \tilde{\mathbf{f}}'\end{bmatrix}\middle|\begin{bmatrix}\mathbf{X}\\ \tilde{\mathbf{X}}\end{bmatrix}\right) = \mathrm{N}\left(\begin{bmatrix}\mathbf{f}\\ \tilde{\mathbf{f}}'\end{bmatrix}\middle|\,\mathbf{0}, \begin{bmatrix}\mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{\mathbf{f},\tilde{\mathbf{f}}'}\\ \mathbf{K}_{\tilde{\mathbf{f}}',\mathbf{f}} & \mathbf{K}_{\tilde{\mathbf{f}}',\tilde{\mathbf{f}}'}\end{bmatrix}\right).$$

The joint posterior for the latent values and the latent value derivatives can be derived from the Bayes' rule

$$p(\mathbf{f}, \tilde{\mathbf{f}}' \,|\, \mathbf{y}, \mathbf{m}, \mathbf{X}, \tilde{\mathbf{X}}) = \frac{p(\mathbf{f}, \tilde{\mathbf{f}}' \,|\, \mathbf{X}, \tilde{\mathbf{X}})p(\mathbf{y} \,|\, \mathbf{f})p(\mathbf{m} \,|\, \tilde{\mathbf{f}}')}{Z}, \tag{5}$$

with $Z = \int p(\mathbf{f}, \tilde{\mathbf{f}}' | \mathbf{X}, \tilde{\mathbf{X}})p(\mathbf{y} \,|\, \mathbf{f})p(\mathbf{m} \,|\, \tilde{\mathbf{f}}')\mathrm{d}\mathbf{f}\mathrm{d}\tilde{\mathbf{f}}'$. Note that since $p(\mathbf{m} \,|\, \tilde{\mathbf{f}}')$ is not Gaussian, the full posterior is analytically intractable and some approximation method must be used. We use expectation propagation (EP) for fast and accurate approximative inference [9].

Model comparison is often done with the energy function, or negative log marginal posterior likelihood of the data $E(\mathbf{y}, \mathbf{m} | \mathbf{X}, \tilde{\mathbf{X}}) = -\log p(\mathbf{y}, \mathbf{m} | \mathbf{X}, \tilde{\mathbf{X}})$. If we are interested in only some part of the model, selected points $\{\mathbf{y}^*, \mathbf{X}^*\}$ can be used to evaluate the model fit

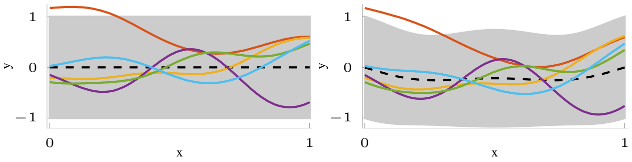
$$E(\mathbf{y}^* | \mathbf{X}^*, \mathbf{y}, \mathbf{m}, \mathbf{X}, \tilde{\mathbf{X}}) = -\log\frac{p(\mathbf{y}^*, \mathbf{y}, \mathbf{m} | \mathbf{X}^*, \mathbf{X}, \tilde{\mathbf{X}})}{p(\mathbf{y}, \mathbf{m} | \mathbf{X}, \tilde{\mathbf{X}})}. \tag{6}$$

3. BAYESIAN OPTIMIZATION WITH DERIVATIVE SIGN OBSERVATIONS

In this section we illustrate how virtual derivative observations can be added to the edges of the search space. In essence, we use the same model as described in Section 2.3 but where the derivative observations are replaced by virtual 'observations' at the boundaries of the domain to correct for the described over-exploration effect.

3.1. Virtual Derivative-Based Search

To encode the prior information that the minimum is not in the boundary set, we propose the following dynamic approach. Just like in the regular BO with GP prior presented in the Section 2, the objective function is given a GP prior which is updated according to the objective function evaluations so far. The next evaluation point is the acquisition function maximum,



(a) (b)

Fig. 3: The GP prior in BO visualized (a) without virtual derivative observations, (b) with virtual derivative observations on borders. Black dotted line is the mean of the GP, the light gray area is 68% central posterior interval and the five lines are random function samples from the prior.

Algorithm 1 Pseudocode of the proposed BO method. The inputs are the acquisition a, the *stopping criterion* and the GP model. Note that this algorithm reduces to standard BO when lines 4-6 are removed.

```
1: while stopping criterion is False do
2:     Fit GP to the available dataset $\mathbf{X}, \mathbf{y}$.
3:     Optimise acquisition function, $a$, to find select new
       location $\mathbf{x}$ to evaluate.
4:     if $\mathbf{x}$ is close to the edge then
5:         Augment $\mathbf{X}$ with a virtual derivative sign observation
           at $\tilde{\mathbf{x}}$.
6:     else
7:         Augment $\mathbf{X}$ with $\mathbf{x}$ and evaluate $g$ at $\mathbf{x}$.
8:     end if
9: end while
```

but if it is closer than threshold ϵ_b to the border of the search space, the point is projected to the border and a virtual derivative observation is placed at that point instead. After having added this virtual observation, the GP posterior is updated and new proposal for the next acquisition is computed. Algorithm 1 contains pseudocode for the proposed method.

As there are no local minima on the borders, the gradient of the function always has a non-zero component pointing outwards from the feasible set. We have no knowledge if there are gradient components in other directions or about the magnitude of the gradient. Thus we only add a component pointing away from the feasible set, $\mathcal{X} \subset \mathcal{R}^d$ and use an observation model that only takes into account if the sign is positive or negative. If the feasible set is a hyper-cube, we can use partial derivatives as observations and thus avoid adding information about other directions. As the control parameter ν in Equation (4) approaches 0 (and $m = 1$), the probit likelihood approaches the unit step function. This means that the likelihood values are close to 1 for all partial derivative values $f' > 0$ and close to zero for all $f' < 0$. Thus we can fix $m_{d_i}^{(i)} = \pm 1$ to positive and negative gradients. The effect of adding virtual derivative observations on the borders of a function is visualized in the Figure 3. From the Figure it can be seen that the virtual derivative observations alter the GP prior to resemble our prior belief of the location of the minimum.

Another parameter to be chosen is the threshold ϵ_b. As acquisitions closer than the threshold value are always rejected, ϵ_b should not be too large. Another argument to avoid too large values is the fading information value of the virtual observations. If the GP allows rapid changes in the latent values, virtual derivative observations affect the posterior distribution only very little. Let l be the diameter of the search space. Our experiments suggest that $\epsilon_b \approx 0.01 \cdot l$ is a good value in most applications.

3.2. Adaptive Search

For some practical applications, we might want to make the presented algorithm more robust to local minima on the border. The following modifications to Algorithm 1 can be used.

Before placing a virtual derivative observation on the border, it can be checked whether or not the existing data supports the virtual gradient sign observation to be added to the model. This can be done by checking the energy values (Equation (6)) of virtual observations of different derivative values, $m_{d_i}^{(i)} \in \{-1, 1\}$.

Since virtual gradient observations only contain information about the sign of the partial derivative, they do not reduce the local variance of a GP similarly as regular observations. As a result, if there are minima on the border, acquisitions might be proposed to locations where virtual observations already exists. If this happens, it is reasonable to remove the virtual observation before adding the new acquisition.

4. EXPERIMENTS

In this section we introduce the four case studies performed to gain insight about the performance of the proposed method. First the details of the experiments are presented and then each study and its results are presented.

4.1. Experimental Set-Up

The proposed Bayesian optimization algorithm was implemented in GPy toolbox[2]. We use zero mean GP prior for regular observations and probit likelihood with $\nu = 10^{-6}$ for the virtual derivative observations and the squared exponential covariance function for the GP.Initial acquisitions are generated with a full factorial design with 2^d points (see 5.3.3.3. from [13]).

Three BO algorithms are used in the case studies. Standard BO algorithm (referred as vanilla BO, VBO), algorithm with virtual derivative sign observations (referred as derivative BO, DBO) and adaptive version of DBO (referred as ADBO). For the last two of these, virtual derivative sign observations are added if the next proposed point is within 1% of the length of the edge of the search space to any border. For ADBO, old virtual derivative observations are removed before adding regular

[2] Toolbox available at: `https://sheffieldml.github.io/GPy/`

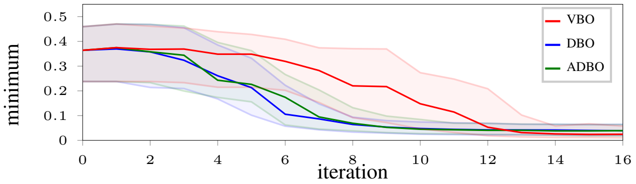


Fig. 4: Median and 25 and 75 percentiles of found minimum of 100 optimization runs as a function of iterations for VBO, DBO and ADBO. Optimization runs are performed for 3 dimensional MND-functions with additive noise of level $s = 0.1$ and LCB as an acquisition function.

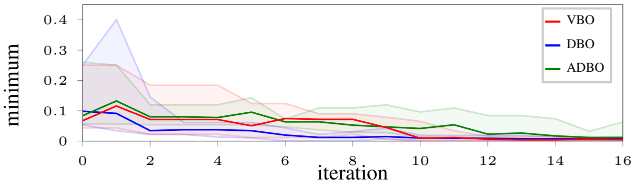


Fig. 5: Same as in Figure 4, but with functions from Sigopt-library.

observation if the Euclidean distance between the points is less than 1% of the length of the edge of the search space.

4.2. Case Study 1: A Simple Example Function

The algorithm is used to illustrate the unwanted boundary over-exploration effect of the regular BO. To show this, a simple function consisting of two Gaussian components is optimized with VBO and DBO using LCB as an acquisition function. The function and 15 first acquisitions are visualized in Figure 2, in the introduction. The results show that VBO over-explores the borders.

4.3. Case Study 2: Random Multivariate Normal Distribution Functions

The algorithms are used to find the minimum of 100 different 3-dimensional multivariate normal distribution (MND) functions where the means and covariances are generated at random. To mimic real life observations, Gaussian noise $\epsilon \sim N(0, 0.1)$ is added to the observations $y(\mathbf{x}) = g(\mathbf{x}) + \epsilon$. 25, 50, and 75 percentiles of found minimum values for the MND functions as a function of iterations are illustrated in Figure 4. The results show that performances of DBO and ADBO are better than or equal to the performance of VBO. It can also be seen that the variance of the optimization performance between different optimization runs is smaller for DBO and ADBO than for VBO.

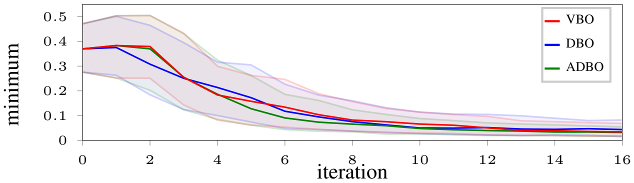


Fig. 6: Same as in Figure 4, but with MND-functions that have local minimum on the edge of the search space.

4.4. Case Study 3: Sigopt Function Library

A benchmark function library[3] Sigopt is developed for evaluating BO algorithms [14]. When taking into account only three dimensional non-discrete functions without local border minima, the library outputs 14 functions. As in the previous case study, to mimic real use cases, the function observations are corrupted with additive Gaussian noise $y(\mathbf{x}) \sim g(\mathbf{x}) + \mathrm{N}(0, 0.1)$. 25, 50, and 75 percentiles of found minimum values of these functions as a function of iterations are illustrated in Figure 5. The results are similar as for MND functions. DBO and ADBO still perform better than VBO. Similarly as before, the variance of the optimization performance between different optimization runs is notably smaller for DBO than for VBO. ADBO performs similarly as VBO. Since there are less functions per dimension, the overall variability in the results is bigger and the percentile curves are not as smooth.

4.5. Case Study 4: Simple Gaussian Functions With Minima on the Border

The algorithms are used to find minimum of similar Gaussian functions as in Section 4.3, with the difference that the global minima of each function is exactly on the border of the search space. The purpose of this case study is to show what happens to the performance of the proposed method if the a priori assumption is violated. 25, 50 and 75 percentiles of found minimum values of these functions as a function of iterations are illustrated in Figure 6 As expected, the results show that DBO does not perform as well as VBO and ADBO. Interestingly DBO performs almost as well as VBO, which shows the robustness of the proposed approach and makes it an appropriate 'default' choice in most problems.

4.6. Case Study 5: Hyper-parameter Optimization of RMSprop

To show the performance for real data, the proposed algorithm was used to tune hyper-parameters of the RMSprop algorithm[4] that used in training a neural network for CIFAR10-data[5]. All

[3] Function library available at: `https://github.com/sigopt/evalset`

[4] RMSprop is an unpublished but established gradient descend method proposed by Geoff Hinton in `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`

[5] Dataset available at: `https://www.cs.toronto.edu/~kriz/cifar.html`

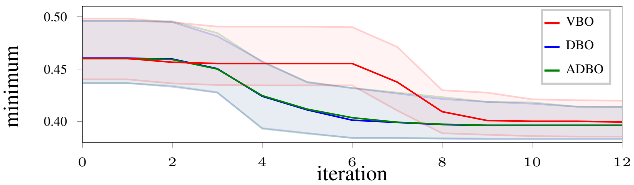


Fig. 7: Same as in Figure 4, but for optimizing hyperparameters of a gradient descend algorithm and without adding noise. Validation error for the found minimum is displayed on y-axis.

the three presented optimizers are used to select the learning rate and decay of the RMSprop-algorithm. Classification error of the validation set as a function of iterations for 100 runs are illustrated in Figure 7. The results show that both the proposed methods perform better than VBO.

5. CONCLUSIONS

We have presented here a Bayesian optimization algorithm which utilizes qualitative prior information concerning the objective function on the borders. Namely, we assume that the gradient of the underlying function points towards the centre on all borders. Typical uses of Bayesian optimization concern expensive functions and in many applications qualitative knowledge of the generic properties of the function are known prior to optimization.

The proposed BO method has proved to significantly improve the optimization speed and the found minimum when comparing the average performance to the performance of the standard BO algorithm without virtual derivative sign observations. The difference in performance is more significant if the assumption of non-existent global or local minima on the border of the search space holds, but is still notable if the assumption is relaxed so that the global minimum is not located on the border.

6. ACKNOWLEDGMENTS

We thank Juho Piironen and Kunal Ghosh for helpful comments to improve the manuscript. We also acknowledge the computational resources of the Aalto Science-IT project.

7. REFERENCES

[1] 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*, vol. 104, no. 1, pp. 148–175, 2016.

[2] B Shahriari, A Bouchard-Côté, and N de Freitas, “Unbounded Bayesian optimization via regularization,” in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 1168–1176.

[3] A Krause and C Guestrin, “Nonmyopic active learning of Gaussian processes: an exploration-exploitation approach,” in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 449–456.

[4] J González, M. A Osborne, and N. D Lawrence, “GLASSES: relieving the myopia of Bayesian optimisation,” in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 790–799.

[5] A O’Hagan, “Some Bayesian numerical analysis,” in *Bayesian statistics 4*, J. M Bernardo, J. O Berger, A. P Dawid, and A. F. M Smith, Eds. 1992, pp. 345–363, Oxford University Press.

[6] O.-P Koistinen, E Maras, A Vehtari, and H Jónsson, “Minimum energy path calculations with Gaussian process regression,” *Nanosystems: Physics, Chemistry, Mathematics*, vol. 7, no. 6, pp. 925–935, 2016.

[7] J Wu, M Poloczek, A. G Wilson, and P. I Frazier, “Bayesian optimization with gradients,” *arXiv preprint arXiv:1703.04389*, 2017.

[8] J. P Gosling, J. E Oakley, and A O’Hagan, “Nonparametric elicitation for heavy-tailed prior distributions,” *Bayesian Analysis*, vol. 2, no. 4, pp. 693–718, 2007.

[9] J. Riihimäki and A Vehtari, “Gaussian processes with monotonicity information.,” in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010, pp. 645–652.

[10] X Wang and J. O Berger, “Estimating shape constrained functions using Gaussian processes,” *SIAM/ASA Journal on Uncertainty Quantification*, vol. 4, no. 1, pp. 1–25, 2016.

[11] N Srinivas, A Krause, S. M Kakade, and M Seeger, “Gaussian process optimization in the bandit setting: No regret and experimental design,” *Proceedings of the 27th International Conference on Machine Learning*, pp. 1015–1022, 2010.

[12] E Solak, S. R Murray, W. E Leithead, D. J Leith, and C. E Rasmussen, “Derivative observations in Gaussian process models of dynamic systems,” in *Advances in Neural Information Processing Systems*, 2003, pp. 1033–1040.

[13] C Carroll, T Paul, and Z Chelli, *Engineering statistics handbook*, NIST iTL, 2013.

[14] I Dewancker, M McCourt, S Clark, P Hayes, A Johnson, and G Ke, “A stratified analysis of Bayesian optimization methods,” *arXiv preprint arXiv:1603.09441*, 2016.