Distributionally Robust Optimization Techniques in Batch Bayesian Optimization

Nikitas Rontsis **Michael A. Osborne** **Paul J. Goulart**
Department of Engineering Science, University of Oxford
{nrontsis, mosb}@robots.ox.ac.uk, paul.goulart@eng.ox.ac.uk

Abstract

We propose a novel, theoretically-grounded, acquisition function for batch Bayesian optimisation informed by insights from distributionally robust optimization. Our acquisition function is a lower bound on the well-known Expected Improvement function – which requires a multi-dimensional Gaussian Expectation over a piecewise affine function – and is computed by evaluating instead the best-case expectation over all probability distributions consistent with the same mean and variance as the original Gaussian distribution. We show that, unlike alternative approaches including Expected Improvement, our proposed acquisition function avoids multi-dimensional integrations entirely, and can be calculated exactly as the solution of a convex optimization problem in the form of a tractable semidefinite program (SDP). Moreover, we prove that the solution of this SDP also yields exact numerical derivatives, which enable efficient optimisation of the acquisition function. Numerical results suggest that our acquisition function performs very similar to the computationally intractable exact Expected Improvement and considerably better than other heuristics.

1 Introduction

When dealing with numerical optimization problems in engineering applications, one is often faced with the optimization of a expensive process that depends on a number of tuning parameters. Examples include the outcome of a biological experiment, training of large scale machine learning algorithms or the outcome of exploratory drilling. We are concerned with problem instances wherein there is the capacity to perform k experiments in *parallel* and, if needed, repeatedly select further batches with cardinality k as part of some sequential decision making process. Given the cost of the process, we wish to select the parameters at each stage of evaluations carefully so as to optimize the process using as few experimental evaluations as possible.

2 Bayesian optimisation

It is common to assume a surrogate model for the outcome $f : \mathbb{R}^n \mapsto \mathbb{R}$ of the process to be optimized. This model, which is built based on both prior assumptions and past function evaluations, is used to determine a collection of k input points for the next set of evaluations. Bayesian Optimization provides an elegant surrogate model approach and has been shown to outperform other state of the art algorithms on a number of challenging benchmark functions [Jones, 2001]. It models the unknown function with a Gaussian Process (GP) [Rasmussen and Williams, 2005], a probabilistic function approximator which can incorporate prior knowledge such as smoothness, trends, etc.

A comprehensive introduction to GPs can be found in [Rasmussen and Williams, 2005]. In short, modeling a function with a GP amounts to modelling the function as a realisation of a stochastic process. In particular, we assume that the outcomes of function evaluations are normally distributed random variables with known *prior* mean function $m : \mathbb{R}^n \mapsto \mathbb{R}$ and *prior* covariance function $\kappa : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$. Prior knowledge for f, such as smoothness and trends, can be incorporated through judicious choice of the covariance function κ, while the mean function m is commonly assumed to be zero. A training dataset $\mathcal{D} = (X^d, y^d)$ of ℓ past function evaluations $y_i^d = f(X_i^d)$ for $i = 1 \dots \ell$, with $y^d \in \mathbb{R}^\ell, X^d \in \mathbb{R}^{\ell \times n}$ is then used to calculate the *posterior* of f.

The GP regression equations [Rasmussen and Williams, 2005] give this posterior $y|\mathcal{D}$ on a batch of k test locations $X \in \mathbb{R}^{k \times n}$ as

$$y|\mathcal{D} \sim \mathcal{N}(\mu(X), \Sigma(X)) \tag{1}$$

with

$$\mu(X) = K(X^d, X)^T K(X^d, X^d)^{-1} y^d, \tag{2}$$

$$\Sigma(X) = K(X, X) - K(X^d, X)^T K(X^d, X^d)^{-1} K(X^d, X) \tag{3}$$

where $K(X, X^d)$ is a $k \times \ell$ matrix containing the prior covariances between every pair of training and test points generated by κ, (similarly for $K(X, X)$). The posterior mean value μ and variance Σ depend also on the dataset $\mathcal{D}$, but we do not explicitly denote their dependence in order to simplify the notation. Likewise, the posterior $y|\mathcal{D}$ is a normally distributed random variable whose mean $\mu(X)$ and covariance $\Sigma(X)$ depend on X, but we will not explicitly indicate this dependence.

Given the surrogate model, we wish to identify some selection criterion for choosing the next batch of points to be evaluated. Such a selection criterion is known as an *acquisition function* in the terminology of Bayesian Optimization. Ideally, such an acquisition function would take into account the number of remaining evaluations that we can afford, e.g. by computing a solution via dynamic programming to construct an optimal sequence of policies for future batch selections. However, a probabilistic treatment of such a criterion is computationally intractable, involving multiple nested minimization-marginalisation steps [González et al., 2016].

To avoid this computational complexity, myopic acquisition functions that only consider the one-step return are typically used instead. For example, one could choose to minimize the one-step *Expected Improvement* (described more fully in §3) over the best evaluation observed so far, or maximize the probability of having an improvement in the next batch over the best evaluation. Other criteria use ideas from bandit [Desautels et al., 2014] and information theory [Shah and Ghahramani, 2015] literatures. In other words, the intractability of the multistep lookahead problem has spurred instead the introduction of a wide variety of myopic heuristics for batch selection.

3 Expected improvement

We will focus on the (one-step) Expected Improvement criterion, which is a standard choice and has been shown to achieve good results in practice [Snoek et al., 2012]. In order to give its formal definition we first require some definitions related to the optimization procedure of the original process. At each step of the optimization define $y^d \in \mathbb{R}^\ell$ as the vector of ℓ past function values evaluated at the points $X^d \in \mathbb{R}^{\ell \times n}$, and $X \in \mathbb{R}^{k \times n}$ as a candidate set of k points for the next batch of evaluations. Then the expected improvement acquisition function is defined as

$$\alpha(X) = \mathbb{E}[\min(y_1, \dots, y_k, \underline{y^d})|\mathcal{D}] - \underline{y^d} \quad \text{with } y|\mathcal{D} \sim \mathcal{N}\big(\mu(X), \Sigma(X)\big) \tag{4}$$

where $\underline{y^d}$ is the element-wise minimum of y^d, i.e. the minimum value of the function f achieved so far by any known function input. Selection of a batch of points to be evaluated with optimal expected improvement then amounts to finding some $X \in \arg\min[\alpha(X)]$.

Unfortunately, direct evaluation of the acquisition function α requires the k–dimensional integration of a piecewise affine function; this is potentially a computationally expensive operation. This is particularly problematic for gradient-based optimization methods, wherein $\alpha(X)$ may be evaluated many times when searching for a minimizer. Regardless of the optimization method used, such a minimizer must also be computed again for every step in the original optimization process, i.e. every time a new batch of points is selected for evaluation. Therefore a tractable acquisition function should be used. In contrast to (4), the acquisition function we will introduce in section 4 avoids expensive integrations, and can be calculated efficiently with standard software tools.

Despite these issues, Chevalier and Ginsbourger [2013] presented an efficient way of approximating α and its derivative $d\alpha/dX$ [Marmin et al., 2015] by decomposing it into a sum of q–dimensional

Gaussian Cumulative Distributions, which can be calculated efficiently using the seminal work of Genz and Bretz [2009]. There are two issues with this approach: First the number of calls to the q–dimensional Gaussian Cumulative Distribution grows quadratically with respect to the batch size q, and secondly, there are no guarantees about the accuracy of the approximation or its gradient. Indeed, approximations of the multi-point expected improvement, as calculated with the `R` package `DiceOptim` [Roustant et al., 2012] can be shown to be arbitrarily wrong in trivial low-dimensional examples (see `https://github.com/oxfordcontrol/Bayesian-Optimization`). To avoid these issues, Gonzalez et al. [2016] and Ginsbourger et al. [2009] rely on heuristics to derive a multi-point criterion. Both methods choose the batch points in a greedy, sequential way, which restricts them from exploiting the interactions between the batch points in a probabilistic manner.

4 Distributionally robust optimisation for Bayesian optimisation

We now proceed to the main contribution of the paper which draws ideas from the Distributionally Robust Optimization community to derive a novel, tractable acquisition function which is a lower bound for the expectation in (4). In particular, we use the posterior result (1)–(3) derived from the GP to determine the mean $\mu(X)$ and variance $\Sigma(X)$ of $y|\mathcal{D}$ given a candidate batch selection X, but we thereafter ignore the Gaussian assumption and consider only that $y|D$ has a distribution embedded within a family of distributions $\mathcal{P}$ that share the mean $\mu(X)$ and covariance $\Sigma(X)$ calculated by (2) and (3). In other words, we define

$$\mathcal{P}(\mu,\Sigma) = \left\{\mathbb{P} \;\middle|\; \mathbb{E}_{\mathbb{P}}[\xi] = \mu, \mathbb{E}_{\mathbb{P}}[\xi\xi^T] = \Sigma\right\}.$$

We will denote the set $\mathcal{P}(\mu(X),\Sigma(X))$ simply as $\mathcal{P}$ where the context is clear. Note in particular that $\mathcal{N}(\mu,\Sigma) \in \mathcal{P}(\mu,\Sigma)$ for any choice of mean μ or covariance Σ.

One can then construct upper and lower bound for the Expected Improvement by maximizing or minimizing over the set $\mathcal{P}$, i.e. by writing

$$\inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[g(\xi)] \le \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}[g(\xi)] \le \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[g(\xi)] \tag{5}$$

where the random vector $\xi \in \mathbb{R}^k$ and the function $g : \mathbb{R}^k \mapsto \mathbb{R}$ are chosen such that $\alpha(X) = \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}[g(\xi)]$, i.e. $\xi = y|\mathcal{D}$ and

$$g(\xi) = \min(\xi_1,\dots,\xi_k,\underline{y^d}) - \underline{y^d}. \tag{6}$$

Observe that the middle term in (5) is equivalent to the expectation in (4).

Perhaps surprisingly, both of the bounds in (5) are computationally tractable even though they seemingly require optimization over the infinite-dimensional (but convex) set of distributions $\mathcal{P}$. For either case, these bounds can be computed exactly via transformation of the problem to a tractable, convex semidefinite optimization problem use distributionally robust optimization techniques [Zymler et al., 2013].

We will focus on the lower bound $\inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[g(\xi)]$ in (5), hence adopting an *optimistic* modelling approach.[1] Informally, we are optimistically assuming that the distribution of function values gives as low a function value as is compatible with the mean and covariance (computed by the GP). For a given mean vector μ and covariance matrix Σ, define the second order moment matrix Ω as

$$\Omega := \begin{bmatrix} \Sigma + \mu\mu^T & \mu \\ \mu^T & 1 \end{bmatrix}, \tag{7}$$

which we will occasionally write as $\Omega(X)$ to highlight the dependency of the second order moment matrix on X. The following result says that the lower (i.e. optimistic) bound in (5) can be computed via the solution of a convex optimization problem whose objective function is linear in Ω:

Theorem 4.1. *The optimal value of the semi-infinite optimization problem*

$$\inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[g(\xi)]$$

[1] It can be shown that the upper bound $\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[g(\xi)]$ in (5) is trivial to evaluate and is not of practical use.

coincides with the optimal value of the following semidefinite program:

$$\begin{aligned} p(\Omega) := \qquad & \textit{sup} \quad \langle \Omega, M \rangle - \underline{y}^d \\ & \textit{subject to} \quad M - C_i \preceq 0, \quad i = 0, \dots, k, \end{aligned} \tag{P}$$

where $M \in \mathbb{S}^{k+1}$ is the decision variable and

$$C_0 := \begin{bmatrix} 0 & 0 \\ 0^T & \underline{y}^d \end{bmatrix}, \quad C_i := \begin{bmatrix} 0 & e_i/2 \\ e_i^T/2 & 0 \end{bmatrix}, \qquad i = 1, \dots, k \tag{8}$$

are auxiliary matrices defined using $\underline{y}^d$ and the standard basis vectors e_i in $\mathbb{R}^k$.

Proof. See Appendix A. □

Problem (P) is a semidefinite program (SDP). SDPs are convex optimization problems with a linear objective and conic constraints, (i.e. constraints over the set of symmetric matrices $\mathbb{S}^k$, positive semidefinite/definite matrices $\mathbb{S}^k_+$/$\mathbb{S}^k_{++}$). Hence it can be solved to global optimality with standard software tools [Sturm, 1999, O'Donoghue et al., 2016]. We therefore propose the computationally tractable acquisition function

$$\bar{\alpha}(X) := p\big(\Omega(X)\big) \leq \alpha(X) \quad \forall X \in \mathbb{R}^{k \times n},$$

which is an optimistic variant of the Expected Improvement function in (4).

Although the value of $p(\Omega)$ for any fixed Ω is computable via solution of an SDP, the complex dependence (2)–(3) that defines the mapping $X \mapsto \Omega(X)$ means that $\bar{\alpha}(X) = p(\Omega(X))$ is still non-convex in X. This is unfortunate, since we ultimately wish to minimize $\bar{\alpha}(X)$ in order to identify the next batch of points to be evaluated experimentally.

However, it is still possible to compute a local solution via local optimisation if an appropriate gradient can be computed. The next result establishes that this is always possible provided that $\Omega \succ 0$:

Theorem 4.2. *$p : \mathbb{S}^{k+1} \mapsto \mathbb{R}$ is differentiable on $\mathbb{S}^{k+1}_{++}$ with $\partial p(\Omega)/\partial\Omega = \bar{M}(\Omega)$, where $\bar{M}(\Omega)$ is the unique optimal solution of (P) at Ω.*

The preceding result shows that $\partial p(\Omega)/\partial\Omega$ is produced as a byproduct of evaluation of $\inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[g(\xi)]$, since it is simply the unique optimizer of (P). Hence we can evaluate $\partial\bar{\alpha}(X)/\partial X$ via $\partial p(\Omega)/\partial\Omega$ using the optimal solution $\bar{M}$ and application of the chain rule, i.e.

$$\frac{\partial \bar{\alpha}(X)}{\partial X_{i,j}} = \left\langle \frac{\partial p(\Omega)}{\partial \Omega}, \frac{\partial \Omega(X)}{\partial X_{(i,j)}} \right\rangle = \left\langle \bar{M}(\Omega), \frac{\partial \Omega(X)}{\partial X_{(i,j)}} \right\rangle \tag{9}$$

Note that the second term in the rightmost inner product in (9) depends on the particular choice of covariance function κ. This computation is standard for many choices of covariance function, and many standard software tools, including `GPy` and `GPflow`, provide the means to compute $\partial\bar{\alpha}(X)/\partial X$ efficiently given $\partial\bar{\alpha}/\partial\Omega = \partial p(\Omega)/\partial\Omega$.

We are now in a position to present an algorithm that summarises our proposed Bayesian Optimisation method using our proposed acquisition function. In the sequel we will present numerical results to demonstrate the validity of this algorithm.

5 Empirical analysis

In this section we demonstrate the effectiveness of our acquisition function against a number of state-of-the-art alternatives. The acquisition functions we consider are shown in Table 1. In the implementation of our own acquisition function, oEI, we produce both acquisition function values and gradients by solving the semidefinite program (P) using the solver MOSEK [MOSEK, 2015] accessed through the Python-based convex optimization package CVXPY [Diamond and Boyd, 2016].

While doing so, extra care should be given in the setup used for the comparison. This is because Bayesian Optimisation is a multifaceted procedure that depends on a collection of disparate elements

```
Algorithm 1: Batch Bayesian Optimisation
Data: Function $f$, dataset of past evaluations $\mathcal{D} = (X^d, y^d)$, covariance function $\kappa$ and mean
      function $m = 0$
Result: A guess for the minimizer $x_{min}$ of $f$
Train $\mathrm{GP}(X^d, y^d)$ by Maximum Likelihood;
while There are remaining function evaluations do
    Choose an initial random batch of points $X_0$;
    // Local gradient-based nonlinear minimization
    $\bar{X} \leftarrow \min(X_0, \texttt{acquisition\_function})$;
    // Append new function evaluations in the dataset
    $X^d \leftarrow \begin{bmatrix} X^d & \bar{X} \end{bmatrix}$;
    $y^d \leftarrow \begin{bmatrix} y^d & f(\bar{X}_1)) \cdots f(\bar{X}_k)) \end{bmatrix}$;
    Retrain $\mathrm{GP}(X^d, y^d)$ by Maximum Likelihood;
end
$x_{min} \leftarrow$ The row of $X^d$ that corresponds to the minimum element of $y_d$;
return $x_{min}$;
Function acquisition_function($X$)
    Calculate $\Omega(X)$ from the GP regression equations (2), (3);
    Solve $(P)$ for $\Omega(X)$;
    $\bar{\alpha} \leftarrow$ optimal value of problem $(P)$;
    $\bar{M} \leftarrow$ optimizer of problem $(P)$;
    return $\bar{\alpha}, \nabla_X \bar{\alpha}(X, \bar{M})$;
```

Key	Description
oEI	Optimistic Expected Improvement *(Our novel algorithm)*
qEI	Multi-point Expected Improvement [Chevalier and Ginsbourger, 2013, Marmin et al., 2015, Roustant et al., 2012]
LP	Local Penalization Expected Improvement [Gonzalez et al., 2016, GPyOpt, 2016]
CL	Constant Liar (max case) [Ginsbourger et al., 2009, Roustant et al., 2012]
EI-Random	Naive batch strategy where the first point is the minimizer of the one-point expected improvement and the rest are chosen uniformly at random
BLCB	Batch Lower Confidence Bound [Desautels et al., 2014]

Table 1: List of acquisition functions used for empirical testing

(e.g. kernel choice, hyperparameter estimation, acquisition function, optimization of the acquisition function) each of which can have a considerable effect on the resulting performance [Snoek et al., 2012]. For this reason we test the different algorithms on a unified testing framework, which is available online at `https://github.com/oxfordcontrol/Bayesian-Optimization`, where the *only* changing element is the acquisition function.

First, we consider one-step scenarios with perfect modelling assumptions, i.e. when the function under minimization is a draw from a GP. At each run a GP is sampled at 10 uniformly random locations at the rectangle $[0, 1]^2$ and then the 2-point expected improvement is calculated to evaluate the choice of each algorithm for a batch size of 2. We average the results for 1000 runs on a GP with a squared exponential kernel [Rasmussen and Williams, 2005] of lengthscale $1/4$. In this setup, qEI obviously yields the best possible results, so we use it as the baseline. An “upper bound” is also

oEI	LP	CL	EI-Random
4.77%	8.45%	9.08%	21%

Table 2: Percentage Difference of the 2-point Expected Improvement achieved via different acquisition functions as compared to the baseline acquisition function (qEI) for 1000 Gaussian Process draws.

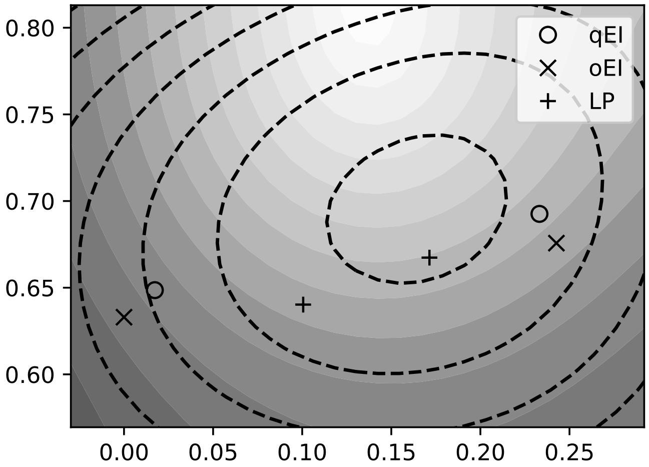

Figure 1: Suggested points using qEI, oEI and LP on a realisation of a GP draw. Dotted lines depict contours of the GP mean, while background color depicts the GP standard deviation. The points suggested by qEI yield the maximum improvement. The points selected by oEI are similar, but are a bit more explorative. On the other hand, LP greedily chooses a point that minimizes the one point return, while the second point is chosen according to a spreading heuristic. As a result, the expected improvement achieved by LP is 14% less than the optimum, while oEI's is within 1% of the optimal value.

calculated, where the first point is chosen according to the 1-point Expected Improvement and the second point is chosen uniformly at random. The results, presented in Table 2, suggest our algorithm as the clear winner, being as much as 1.9x closer to the baseline as other heuristics. This is because both CL and LP are based on a greedy sequential selection strategy, the effects of which are depicted in Figure 1.

Having shown that oEI provides the closest performance to qEI amongst the acquisition functions of Table 1 that approximate qEI we next evaluate its performance in minimizing synthetic benchmark functions. We consider a mixture of cosines defined in $[0,1]^2$, the Branin-Hoo function, defined on $[-5,10] \times [1,15]$, the Six-Hump Camel function defined on $[-2,2] \times [-1,1]$ and the Alpine-1 function defined on $[-10,10]^5$. We compare the performance of oEI against qEI, BLCB, as well as random uniform sampling. The initial dataset consists of 10 points at random locations, and we run a Bayesian optimization loop of 10 iterations of batch size 5. A squared exponential kernel with automatic relevance determination is used for the GP modelling [Rasmussen and Williams, 2005]. Its hyperparameters are estimated by maximizing the likelihood via local optimisation with 20 random restarts. Likewise, the acquisition functions are optimized via local optimisation with 30 random restarts. The standard quasi-newton L-BFGS-B algorithm [Fletcher, 1987] is used to perform local optimisation. Error bars are presented for 40 independent runs with different initial dataset. All of the methods tested managed to identify the global minimum of the three two-dimensional functions (Cosines, Branin-Hoo, Six-Hump Camel). The performance of qEI is close to oEI, which, again, supports the claim that oEI is a promising tractable alternative to qEI. On the other hard, the transient performance of BLCB is considerably different. This is expected, as BLCB does not try to minimize the one-step return, but the return over the course of the optimization [Desautels et al., 2014]. Lastly,

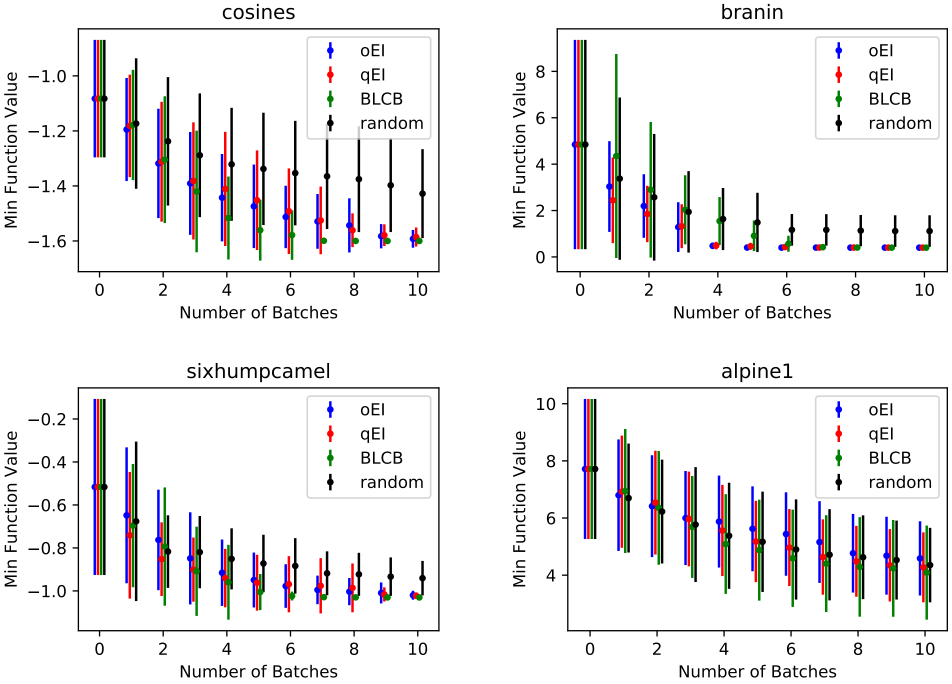

Figure 2: Results of optimizing synthetic benchmark functions with an initial dataset of 10 points over 10 iterations of batch size 5. Dots depict mean performance over 40 random runs, while lines depict confidence intervals (± sd)

all of the algorithms exhibit poor performance, equivalent to random sampling, on the much more challenging 5-d Alpine-1 function, which suggests that the underling GP fails to model this function.

6 Conclusions

We introduced a new acquisition function that is a tractable, probabilistic relaxation of the multi point Expected Improvement, drawing ideas from the Distributionally Robust Optimization community. It can be calculated exactly as the solution of a Semidefinite Program, with its gradient calculated as a direct byproduct of the solution. Unlike competitors, the suggested acquisition functions scales well with batch size, does not rely on heuristics and directly considers the correlation between the suggested points, achieving performance close to the actual multi-point Expected Improvement while remaining computationally tractable. Future work will try to bring even more results from the Optimization community: sensitivity results on Semidefinite Programming [Freund and Jarre, 2004] can provide the Hessian of the acquisition function allowing a Newton-based method to be used in the challenging problem of minimizing the high-dimensional acquisition function.

Acknowledgments

This work was supported by the EPSRC AIMS CDT grant EP/L015987/1 and Schlumberger. Many thanks to L. Berrada for the helpful discussions.

References

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Clément Chevalier and David Ginsbourger. *Fast Computation of the Multi-Points Expected Improvement with Applications in Batch Selection*, pages 59–69. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-44973-4. URL `http://dx.doi.org/10.1007/978-3-642-44973-4_7`.

Thomas Desautels, Andreas Krause, and Joel W. Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15:4053–4103, 2014. URL `http://jmlr.org/papers/v15/desautels14a.html`.

Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

R. Fletcher. *Practical Methods of Optimization; (Second Edition)*. Wiley-Interscience, New York, NY, USA, 1987. ISBN 0-471-91547-5.

Roland W. Freund and Florian Jarre. A sensitivity result for semidefinite programs. *Oper. Res. Lett.*, 32(2): 126–132, March 2004. ISSN 0167-6377. doi: 10.1016/S0167-6377(03)00069-5. URL `http://dx.doi.org/10.1016/S0167-6377(03)00069-5`.

Alan Genz and Frank Bretz. *Computation of Multivariate Normal and T Probabilities*. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 364201688X, 9783642016882.

David Ginsbourger, Rodolphe Le Riche, Laurent Carraro, and Département Mi. A multi-points criterion for deterministic parallel global optimization based on gaussian processes. *Journal of Global Optimization, in revision*, 2009.

D. Goldfarb and K. Scheinberg. On parametric semidefinite programming. *Applied Numerical Mathematics*, 29 (3):361 – 377, 1999. ISSN 0168-9274. URL `http://dx.doi.org/10.1016/S0168-9274(98)00102-0`.

Javier Gonzalez, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch bayesian optimization via local penalization. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 648–657, Cadiz, Spain, 09–11 May 2016. PMLR. URL `http://proceedings.mlr.press/v51/gonzalez16a.html`.

Javier González, Michael A. Osborne, and Neil D. Lawrence. GLASSES : Relieving The Myopia Of Bayesian Optimisation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016. URL `http://arxiv.org/pdf/1510.06299v1.pdf`.

GPyOpt. A bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`, 2016.

Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001. ISSN 1573-2916. URL `http://dx.doi.org/10.1023/A:1012771025575`.

Sébastien Marmin, Clément Chevalier, and David Ginsbourger. *Differentiating the Multipoint Expected Improvement for Optimal Batch Design*, pages 37–48. Springer International Publishing, Cham, 2015. ISBN 978-3-319-27926-8. URL `http://dx.doi.org/10.1007/978-3-319-27926-8_4`.

ApS MOSEK. The MOSEK optimization toolbox for Python manual. *Version 7.1 (Revision 28)*, 2015. URL `http://www.mosek.com/`.

B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, June 2016. URL `http://stanford.edu/~boyd/papers/scs.html`.

Motakuri V Ramana, Levent Tunçel, and Henry Wolkowicz. Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7(3):641–662, 1997.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

Olivier Roustant, David Ginsbourger, Yves Deville, et al. Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(i01), 2012.

Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3330–3338. Curran Associates, Inc., 2015.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Version 1.05 available from `http://fewcal.kub.nl/sturm`.

Steve Zymler, Daniel Kuhn, and Berç Rustem. Distributionally robust joint chance constraints with second-order moment information. *Mathematical Programming*, 137(1):167–198, 2013. ISSN 1436-4646. URL `http://dx.doi.org/10.1007/s10107-011-0494-7`.

Distributionally Robust Optimization Techniques in Batch Bayesian Optimization

Supplemental Proofs

A Value of the Expected Improvement Lower Bound

In this section we provide a proof of the first of our main results, Theorem 4.1, which establishes that one can compute the value of our optimistic lower bound function

$$\inf_{\mathbb{P}\in\mathcal{P}(\mu,\Sigma)} \mathbb{E}_{\mathbb{P}}[g(\xi)] \tag{10}$$

via solution of a convex optimization problem in the form of a semidefinite program.

Proof of Theorem 4.1:

Recall that the set $\mathcal{P}(\mu,\Sigma)$ is the set of all distributions with mean μ and covariance Σ. Following the approach of Zymler et al. [2013], we first remodel problem (10) as:

$$\begin{aligned} \inf_{\nu\in\mathcal{M}_+} \quad & \int_{\mathbb{R}^k} g(\xi)\nu(d\xi) \\ \text{subject to} \quad & \int_{\mathbb{R}^k} \nu(d\xi) = 1 \\ & \int_{\mathbb{R}^k} \xi\nu(d\xi) = \mu \\ & \int_{\mathbb{R}^k} \xi\xi^T\nu(d\xi) = \Sigma + \mu\mu^T, \end{aligned} \tag{11}$$

where $\mathcal{M}_+$ represents the cone of nonnegative Borel measures on $\mathbb{R}^k$. The optimization problem (11) is a semi-infinite linear program, with infinite dimensional decision variable ν and a finite collection of linear equalities in the form of moment constraints.

As shown by Zymler et al. [2013], the dual of problem (11) has instead a finite dimensional set of decision variables and an infinite collection of constraints, and can be written as

$$\begin{aligned} \text{maximize} \quad & \langle\Omega, M\rangle \\ \text{subject to} \quad & \big[\xi^T\ 1\big]M\big[\xi^T\ 1\big]^T \le g(\xi) \quad \forall\xi\in\mathbb{R}^k, \end{aligned} \tag{12}$$

with $M\in\mathbb{S}^{k+1}$ the decision variable and $\Omega\in\mathbb{S}^{k+1}$ the second order moment matrix of ξ (see (7)). Strong duality holds between problems (11) and (12), i.e. there is zero duality gap and their optimal values coincide.

The dual decision variables in (12) form a matrix M of Lagrange multipliers for the constraints in (11) that is block decomposable as

$$M = \begin{pmatrix} M_{11} & m_{12} \\ m_{12}^T & m_{22} \end{pmatrix},$$

where $M_{11}\in\mathbb{R}^{k\times k}$ are multipliers for the second moment constraint, $m_{12}\in\mathbb{R}^k$ multipliers for the mean value constraint, and m_{22} a scalar multiplier for the constraint that $\nu\in\mathcal{M}_+$ should integrate to 1, i.e. that ν should be a probability measure.

For our particular problem, we have $g(\xi) = \min(\xi_{(1)},\ldots,\xi_{(k)},\underline{y^d}) - \underline{y^d}$ as defined in (6), so that (12) can be rewritten as

$$\begin{aligned} \text{maximize} \quad & \langle\Omega, M\rangle - \underline{y^d} \\ \text{subject to} \quad & \big[\xi^T\ 1\big]M\big[\xi^T\ 1\big]^T \le \xi_{(i)} \qquad \forall\xi\in\mathbb{R}^k, \quad i=1,\ldots,k \\ & \big[\xi^T\ 1\big]M\big[\xi^T\ 1\big]^T \le \underline{y^d}. \end{aligned} \tag{13}$$

The infinite dimensional constraints in (13) can be replaced by the equivalent conic constraints

$$M - \begin{bmatrix} 0 & e_i/2 \\ e_i^T/2 & 0 \end{bmatrix} \preceq 0, \qquad i = 1, \dots, k$$

and

$$M - \begin{bmatrix} 0 & 0 \\ 0^T & \underline{y}^d \end{bmatrix} \preceq 0,$$

respectively, where e_i are the standard basis vectors in $\mathbb{R}^k$. Substituting the above constraints in (13) results in (P), which proves the claim. □

B Gradient of the Expected Improvement Lower Bound

In this section we provide a proof of our second main result, Theorem 4.2, which shows that the gradient $\partial p / \partial \Omega$ of our lower bound function (5) with respect to Ω coincides with the optimal solution of the semidefinite program (P).

Before proving Theorem 4.2 we require two ancillary results. The first of these results establishes that any feasible point M for the optimization problem (P) has strictly negative definite principal minors in the upper left hand corner.

Lemma B.1. *For any feasible $M \in \mathbb{S}^{k+1}$ of (P) the upper left $k \times k$ matrix M_{11} is negative definite.*

Proof. Let

$$M = \begin{bmatrix} M_{11} & m_{12} \\ m_{12}^T & m_{22} \end{bmatrix}$$

where $M_{11} \in \mathbb{S}^k, m_{12} \in \mathbb{R}^k$ and $m_{22} \in \mathbb{R}$.

Using the definitions of C_i for $i = 0, \dots, k$ from (8), the semidefinite constraint in the optimization problem (P) can be expressed as

$$\forall \big(x \in \mathbb{R}^k, z \in \mathbb{R}\big) \begin{cases} x^T M_{11} x + 2 m_{12}^T x z + (m_{22} - \underline{y}^d) z^2 \quad \leq 0 & (14) \\ x^T M_{11} x + 2(m_{12} - \dfrac{e_i}{2})^T x z + m_{22} z^2 \leq 0 \quad i = 1, \dots, k & (15) \end{cases}$$

We can easily see that $M_{11} \preceq 0$ by substituting $z = 0$ in (15). It remains to show that this matrix is actually sign definite, which amounts to showing that it is full rank.

Assume the contrary, so that there exists some nonzero $x \in \mathbb{R}^k$ satisfying $x^T M_{11} x = 0$. Then, (15) gives $2(m_{12} - e_i/2)^T x z + m_{22} z^2 \leq 0$. When $(m_{12} - e_i/2)^T x \neq 0$ for some i, a sufficiently small z can be chosen such that $(2(m_{12} - e_i/2)^T x z) > 0$ and the constraint (15) is violated.

On the other hand, when $(m_{12} - e_i/2)^T x = 0$ we can conclude that $m_{12}^T x - x_i/2 = 0 \; \forall i = 1, \dots, k$, which means that $x = x_0 \mathbf{1} \neq 0$ where $x_0 \in \mathbb{R}$ and $\mathbf{1}^T m_{12} = 1/2$. In that case (14) is violated for a sufficiently small positive z. Hence M_{11} is full rank by contradiction and $M_{11} \prec 0$. □

Our second ancillary result considers the gradient of the function p when its argument is varied linearly along some direction $\bar{\Omega}$.

Lemma B.2. *Given any $\bar{\Omega} \in \mathbb{S}^{k+1}$ and any moment matrix $\Omega \in \mathbb{S}_{++}^{k+1}$, define the scalar function $q(\cdot\,; \Omega) : \mathbb{R} \to \mathbb{R}$ as*

$$q(\gamma; \Omega) := p(\Omega + \gamma \bar{\Omega}).$$

Then $q(\cdot\,; \Omega)$ is differentiable at 0 with $\partial q(\gamma; \Omega)/\partial \gamma|_{\gamma=0} = \langle \bar{\Omega}, \bar{M}(\Omega) \rangle$, where $\bar{M}(\Omega)$ is the optimal solution of (P) at Ω.

Proof. Define the set Γ_Ω as

$$\Gamma_\Omega := \{\gamma \mid \gamma \in \operatorname{dom} q(\cdot\,; \Omega)\} = \big\{\gamma \;\big|\; \big(\Omega + \gamma \bar{\Omega}\big) \in \operatorname{dom} p\big\},$$

i.e. the set of all γ for which problem (P) has a bounded solution given the moment matrix $\Omega + \gamma \bar{\Omega}$. In order to prove the result it is then sufficient to show:

i) $0 \in \operatorname{int} \Gamma_\Omega$, and

ii) The solution of (P) at Ω is unique.

The remainder of the proof then follows from [Goldfarb and Scheinberg, 1999, Lemma 3.3], wherein it is shown that $0 \in \operatorname{int} \Gamma_\Omega$ implies that $\langle \bar{\Omega}, \bar{M}(\Omega) \rangle$ is a subgradient of $q(\cdot\,; \Omega)$ at 0, and subsequent remarks in Goldfarb and Scheinberg [1999] establish that uniqueness of the solution $M(\Omega)$ ensure differentiability.

We will show that both of the conditions (i) and (ii) above are satisfied. The proof relies on the Lagrange dual of problem (P), which we can write as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=0}^{k} \langle Y_i, C_i \rangle - \underline{y^d} \\ \text{subject to} \quad & Y_i \succeq 0, \quad i = 0, \ldots, k \\ & \sum_{i=0}^{k} Y_i = \Omega. \end{aligned} \tag{D}$$

(i): Proof that $0 \in \operatorname{int} \Gamma_\Omega$*:*

It is well-known that if both of the primal and dual problems (P) and (D) are strictly feasible then their optimal values coincide, i.e. Slater's condition holds and we obtain strong duality; see [Boyd and Vandenberghe, 2004, Section 5.2.3] and [Ramana et al., 1997].

For (P) it is obvious that one can construct a strictly feasible point. For (D), $Y_i = \Omega/(k+1)$ defines a strictly feasible point for any $\Omega \succ 0$. Hence (P) is solvable for any $\Omega + \gamma\bar{\Omega}$ with γ sufficiently small. As a result, $0 \in \operatorname{int} \Gamma$.

(ii): Proof that the solution to (P) *at* Ω *is unique:*

We can prove uniqueness by examining the Karush-Kuhn-Tucker conditions for a pair of primal and dual solutions $\bar{M}$, $\{\bar{Y}_i\}$:

$$C_i - \bar{M} \succeq 0 \tag{16}$$

$$\bar{Y}_i \succeq 0 \tag{17}$$

$$\langle \bar{Y}_i, \bar{M} - C_i \rangle = 0 \Rightarrow \bar{Y}_i(\bar{M} - C_i) = 0 \tag{18}$$

$$\left. \frac{\partial \mathcal{L}(M, \Omega)}{\partial M} \right|_{\bar{M}} = 0 \Rightarrow \sum_{i=0}^{k} \bar{Y}_i = \Omega, \tag{19}$$

where $\mathcal{L}$ denotes the Lagrangian of (P). As noted before, such a pair of solutions exists when $\Omega \succ 0$.

First, we will prove that $\operatorname{rank}(\bar{M} - C_i) = k$ and $\operatorname{rank}(\bar{Y}_i) = 1,\ \forall i = 0, \ldots, k$. Lemma B.1 implies that $[x^T 0](\bar{M} - C_i)[x^T 0]^T = [x^T 0]\bar{M}[x^T 0]^T < 0, \forall x \in \mathbb{R}^k$ (recall that C_i is nonzero only in the last column or the last row), which means that $\operatorname{rank}(\bar{M} - C_i) \geq k$. Due to the complementary slackness condition (18), the span of $\bar{Y}_i$ is orthogonal to the span of $\bar{M} - C_i$ and consequently $\operatorname{rank}(\bar{Y}_i) \leq 1$. However, according (19) to we have

$$\operatorname{rank} \sum_{i=0}^{k} \bar{Y}_i = \operatorname{rank}(\Omega) \overset{\Omega \succ 0}{\Longrightarrow} \sum_{i=0}^{k} \operatorname{rank}(\bar{Y}_i) \geq k + 1 \tag{20}$$

which results in

$$\operatorname{rank}(\bar{M} - C_i) = k, \quad \operatorname{rank}(\bar{Y}_i) = 1, \tag{21}$$

with

$$\mathcal{R}(\bar{Y}_i) = \mathcal{N}(\bar{M} - C_i), \quad i = 0, \ldots, k \tag{22}$$

where the final equality uses (18), and where $\mathcal{N}(\cdot)$ and $\mathcal{R}(\cdot)$ denote the kernel and image of a matrix, respectively.

Since for every $i = 0, \dots, k$ the kernel $\mathcal{N}(\bar{M} - C_i)$ is of dimension one, we can uniquely identify (up to a sign change) a vector n_i in the kernel with unit norm. Moreover, according to (21) we can represent every $\bar{Y}_i$ as

$$\bar{Y}_i = \lambda_i n_i n_i^T, \quad \forall i = 0, \dots, k. \tag{23}$$

for some $\lambda_i \in \mathbb{R}_{++}, \ i = 0, \dots, k$. The positivity of λ_i comes from the fact that $\bar{Y}_i \succeq 0$ is of rank one.

We next show that the set of null vectors $\{n_i\}$ are the same (up to a sign change) for every primal-dual solution. Assume that $\bar{M} + \delta M$, with $\delta M \in \mathbb{S}^{k+1}$ is also optimal and $\{\tilde{n}_i\}$ are the unit norm null vectors of $\{\bar{M} + \delta M - C_i\}$. By definition we have

$$\tilde{n}_i^T(\delta M + \bar{M} - C_i)\tilde{n}_i = 0 \quad i = 0, \dots, k. \tag{24}$$

However, since $\bar{M}$ is feasible we have $\tilde{n}_i^T(\bar{M} - C_i)\tilde{n}_i \le 0$, which results in

$$\tilde{n}_i^T \delta M \tilde{n}_i \ge 0, \quad i = 0, \dots, k. \tag{25}$$

Since $\bar{M}$ and $\bar{M} + \delta M$ have the same objective value we conclude that $\langle \Omega, \delta M \rangle = 0$. Moreover, according to (19) and (23) we have the conditions $\Omega = \sum_{i=0}^k \bar{Y}_i = \sum_{i=0}^k \tilde{\lambda}_i \tilde{n}_i \tilde{n}_i^T$ for some $\tilde{\lambda}_i \in \mathbb{R}_{++}$. Hence

$$\operatorname{tr}(\Omega \delta M) = 0 \Rightarrow \operatorname{tr}(\delta M \sum_{i=0}^k \tilde{\lambda}_i \tilde{n}_i \tilde{n}_i^T) = 0$$

for some $\tilde{\lambda}_i > 0$, which in turn implies that

$$\begin{aligned}
& \sum_{i=0}^k \tilde{\lambda}_i \operatorname{tr}(\delta M \tilde{n}_i \tilde{n}_i^T) && = 0 \\
\Longrightarrow \quad & \sum_{i=0}^k \tilde{\lambda}_i \tilde{n}_i^T \delta M \tilde{n}_i && = 0 \\
\overset{(25)}{\Longrightarrow} \quad & \tilde{\lambda}_i \tilde{n}_i^T \delta M \tilde{n}_i && = 0 \quad \forall i = 0, \dots, k \\
\overset{(24)}{\Longrightarrow} \quad & \tilde{n}_i(\bar{M} - C_i)\tilde{n}_i^T && = 0 \quad \forall i = 0, \dots, k.
\end{aligned}$$

Since $\bar{M} - C_i \preceq 0$, this proves that n_i is the uniquely defined unit norm null vector (up to a change in sign) of $\bar{M} - C_i$. This proves that $n_i = \tilde{n}_i, i = 0, \dots, k$.

Given the null vectors $\{n_i\}$, the scalar values $\{\lambda_i\}$ can be uniquely defined using (19), as they are the coefficients of the decomposition of the rank $k+1$ matrix Ω to the $k+1$ rank 1 matrices $\{\bar{Y}_i\}$. Thus, given the uniqueness of $\{n_i\}$ across every solution $\bar{M} + \delta M$, (23) proves uniqueness of $\{\bar{Y}_i\}$, i.e. uniqueness of the dual solution. Now we can easily prove uniqueness for the primal solution. Summing (18) gives

$$\sum_{i=0}^k \bar{Y}_i(\bar{M} - C_i) = 0 \overset{(19)}{\Leftrightarrow} \Omega \bar{M} = \sum_{i=0}^k \bar{Y}_i C_i \Leftrightarrow \bar{M} = \Omega^{-1} \sum_{i=0}^k \bar{Y}_i C_i. \qquad \square$$

Proof of Theorem 4.2:

Given the preceding support results of this section, we are now in a position to prove Theorem 4.2.

We begin by considering the derivative of the solution of (P) when perturbing Ω across a specific direction $\bar{\Omega}$, i.e. $\partial q(\gamma; \Omega)/\partial \gamma$ with $q(\gamma; \Omega) = p(\Omega + \gamma \bar{\Omega})$. Lemma B.2 shows that $\partial q(\gamma; \Omega)/\partial \gamma|_0 = \langle \bar{\Omega}, \bar{M} \rangle$ when $\Omega \succ 0$. The proof then follows element-wise from Lemma B.2 by choosing $\bar{\Omega}$ a sparse matrix with $\bar{\Omega}_{(i,j)} = 1$ the only nonzero element. $\square$