GLASSES: Relieving The Myopia Of Bayesian Optimisation

Anonymous Author 1
Unknown Institution 1

Anonymous Author 2
Unknown Institution 2

Anonymous Author 3
Unknown Institution 3

Abstract

We present GLASSES: Global optimisation with Look-Ahead through Stochastic Simulation and Expected-loss Search. The majority of global optimisation approaches in use are myopic, in only considering the impact of the next function value; the non-myopic approaches that do exist are able to consider only a handful of future evaluations. Our novel algorithm, GLASSES, permits the consideration of dozens of evaluations into the future. We show that the far-horizon planning thus enabled leads to substantive performance gains in empirical tests.

1 Introduction

Global optimisation is core to any complex problem where design and choice play a role. Within Machine Learning, such problems are found in the tuning of hyperparameters [Snoek et al., 2012], sensor selection [Garnett et al., 2010] or experimental design [Martinez-Cantin et al., 2009]. Most global optimisation techniques are myopic, in considering no more than a single step into the future. Relieving this myopia requires solving the *multi-step lookahead* problem: the global optimisation of an function by considering the significance of the next function evaluation on function evaluations (steps) further into the future. It is clear that a solution to the problem would offer performance gains. For example, consider the case in which we have a budget of two evaluations with which to optimise a function $f(x)$ over the domain $\mathcal{X} = [0, 1] \subset \mathbb{R}$. If we are strictly myopic, our first evaluation will likely be at $x = 1/2$, and our second then at only one of $x = 1/4$ and $x = 3/4$. This myopic strategy will thereby result in ignoring half of the domain $\mathcal{X}$, regardless of the second choice. If we adopt a two-step lookahead approach, we will select function evaluations that will be more evenly distributed across the domain by the time the budget is exhausted. We will consequently be better informed about f and its optimum.

There is a limited literature on the multi-step lookahead problem. Osborne et al. [2009] perform multi-step lookahead by optimising future evaluation locations, and sampling over future function values. This approach scales poorly with the number of future evaluations considered, and the authors present results for no more than two-step lookahead. [Marchant et al., 2014] reframe the multi-step lookahead problem as a partially observed Markov decision process, and adopt a Monte Carlo tree search approach in solving it. Again, the scaling of the approach permits the authors to consider no more than six steps into the future. In the past, the multi-step look ahead problem was studied by Streltsov and Vakili [1999] proposing a utility function that is provably a globally optimal in cases where the model of the function values remains unchanged.

Interestingly, there exists a link between the multi-step lookahead problem and *batch* Bayesian optimisation [Ginsbourger et al., 2009, Azimi et al., 2011, 2012]. In this later case, batches of locations rather than individual observations are selected in each iteration of the algorithm and evaluated in parallel. When such locations are selected *greedily*, that is, one after the other, the key to selecting good batches relies on the ability of the batch criterion of predicting future steps of the algorithm. In this work we will exploit this parallelism to compute a non-myopic loss for Bayesian optimisation. Our algorithm, GLASSES, provides scaling superior to existing alternatives, which is tested in a variety of experiments. In Section 2 we formalise the problem and describe the contributions of this work. Section 3 describe the details of the proposed algorithm. Section 4 illustrate the superior performance of GLASSES and we conclude in Section 5 with a discussion about the most interesting results observed in this work.

Preliminary work. Under review by AISTATS 2016. Do not distribute.

2 Background and challenge

2.1 Bayesian Optimisation with one step look-ahead

Let $f : \mathcal{X} \to \mathbb{R}$ be well behaved function defined on a compact subset $\mathcal{X} \subseteq \mathbb{R}^d$. We are interested in solving the global optimization problem of finding

$$\mathbf{x}_M = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}).$$

We assume that f is a *black-box* from which only perturbed evaluations of the type $y_i = f(\mathbf{x}_i) + \epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, are available. Bayesian Optimization is an heuristic strategy to make a series of evaluations $\mathbf{x}_1, \dots, \mathbf{x}_n$ of f, typically very limited in number, such that the the minimum of f is evaluated as soon as possible [Lizotte, 2008, Jones, 2001, Snoek et al., 2012, Brochu et al., 2010].

Assume that n points have been gathered so far, having a dataset $\mathcal{D}_0 = \{(\mathbf{x}_i, y_i)\}_{i=1}^N = (\mathbf{X}_0, \mathbf{Y}_0)$. Before collecting any new point, a surrogate probabilistic model for f is calculated. This is topically a Gaussian Process (GP) $p(f) = \mathcal{GP}(\mu; k)$ with mean function μ and a covariance function k, and whose parameters will be denoted by θ. Let $\mathcal{I}_0$ be the current available information: the conjunction of $\mathcal{D}_0$, the model parameters and the model likelihood type. Under Gaussian likelihoods, the predictive distribution for y_* at $\mathbf{x}_*$ is also Gaussian with mean posterior mean and variance

$$\mu(\mathbf{x}_*|\mathcal{I}_0) = \mathbf{k}_\theta(\mathbf{x}_*)^\top[\mathbf{K}_\theta + \sigma^2\mathbf{I}]^{-1}\mathbf{y} \text{ and}$$

$$\sigma^2(\mathbf{x}_*|\mathcal{I}_0) = k_\theta(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_\theta(\mathbf{x}_*)^\top[\mathbf{K}_\theta + \sigma^2\mathbf{I}]^{-1}\mathbf{k}_\theta(\mathbf{x}_*),$$

where $\mathbf{K}_\theta$ is the matrix such that $(\mathbf{K}_\theta)_{ij} = k_\theta(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}_\theta(\mathbf{x}_*) = [k_\theta(\mathbf{x}_1, \mathbf{x}_*), \dots, k_\theta(\mathbf{x}_n, \mathbf{x}_*)]^\top$ [Rasmussen and Williams, 2005].

Given the GP model, we now need to determine the best location to sample. Imagine that we only have one remaining evaluation ($n = 1$) before we need to report our inferred location about the minimum of f. Denote by $\eta = \min\{\mathbf{Y}_0\}$, the current best found value. We can define the loss of evaluating f this last time at $\mathbf{x}_*$ assuming it is returning y_* as

$$\lambda(y_*) \triangleq \begin{cases} y_*; & \text{if} \quad y_* \leq \eta \\ \eta; & \text{if} \quad y_* > \eta. \end{cases}$$

Therefore the loss corresponds is the new observed minimum, $\min(\eta, y_*)$. Its expectation is

$$\Lambda_1(\mathbf{x}_*|\mathcal{I}_0) \triangleq \mathbb{E}[\min(y_*, \eta)] = \int \lambda(y_*)p(y_*|\mathbf{x}_*, \mathcal{I}_0)\mathrm{d}y_*$$

where the subscript in Λ refers to the fact that we are considering one future evaluations. Giving the properties of the GP, $\Lambda_1(\mathbf{x}_*|\mathcal{I}_0)$ can be computed in closed form for any $\mathbf{x}_* \in \mathcal{X}$. In particular, for Φ the usual Gaussian cumulative distribution function, we have that

$$\begin{aligned} \Lambda_1(\mathbf{x}_*|\mathcal{I}_0) &\triangleq \eta\int_\eta^\infty \mathcal{N}(y_*; \mu, \sigma^2)\mathrm{d}y_* \\ &+ \int_{-\infty}^\eta y_*\mathcal{N}(y_*; \mu, \sigma^2)\mathrm{d}y_* \\ &= \eta + (\mu - \eta)\Phi(\eta; \mu, \sigma^2) - \sigma^2\mathcal{N}(\eta, \mu, \sigma^2), \end{aligned} \quad (1)$$

where we have abbreviated $\sigma^2(y_*|\mathcal{I}_0)$ as σ^2 and $\mu(y_*|\mathcal{I}_0)$ as μ. Finally, the next evaluation is located where $\Lambda_1(\mathbf{x}_*|\mathcal{I}_0)$ gives the minimum value. This point can be obtained by any gradient descent algorithm since analytical expressions for the gradient and Hessian of $\Lambda_1(\mathbf{x}_*|\mathcal{I}_0)$ exist [Osborne, 2010].

2.2 Looking many steps ahead

Expression (1) can also be used as a myopic approximation to the optimal decision when n evaluations of f remain available. Indeed, most BO methods are myopic and ignore the future decisions that will be made by the algorithm in the future steps.

Denote by $\{(\mathbf{x}_j, y_j)\}$ for $j = 1, \dots, n$ the remaining n available evaluations and by $\mathcal{I}_j$ the available information after the data set $\mathcal{D}_0$ has been augmented with $(\mathbf{x}_j, y_j), \dots, (\mathbf{x}_j, y_j)$ and the parameters θ of the model updated. We use $\Lambda_n(\mathbf{x}_*|\mathcal{I}_0)$ to denote the expected loss of selecting $\mathbf{x}_*$ given $\mathcal{I}_0$ and considering n future evaluations. A Proper Bayesian formulation allows us to define this *long-sight* loss [Osborne, 2010] as

$$\begin{aligned} \Lambda_n(\mathbf{x}_*|\mathcal{I}_0) &= \int \lambda(y_n)\prod_{j=1}^n p(y_j|\mathbf{x}_j, \mathcal{I}_{j-1})p(\mathbf{x}_j|\mathcal{I}_{j-1}) \\ &\quad \mathrm{d}y_* \dots \mathrm{d}y_n\mathrm{d}\mathbf{x}_2 \dots \mathrm{d}\mathbf{x}_n{}^1 \end{aligned} \quad (2)$$

where

$$p(y_j|\mathbf{x}_j, \mathcal{I}_{j-1}) = \mathcal{N}\left(y_j; \mu(\mathbf{x}_j; \mathcal{I}_{j-1}), \sigma^2(\mathbf{x}_j|\mathcal{I}_{j-1})\right)$$

is the predictive distribution of the GP at $\mathbf{x}_j$ and

$$p(\mathbf{x}_j|\mathcal{I}_{j-1}) = \delta(\mathbf{x}_j - \arg\min_{\mathbf{x}_*\in\mathcal{X}} \Lambda_{n-j+1}(\mathbf{x}_*|\mathcal{I}_{j-1}))$$

reflects the optimization step required to obtain $\mathbf{x}_j$ after all previous the evaluations f have been iteratively optimized and marginalized. The graphical probabilistic model underlying (2) is illustrated in Figure 1.

To evaluate Eq. (2) we can successively sample from y_1 to y_{j-1} and optimize for the appropriate $\Lambda_{n-j+1}(\mathbf{x}_*|\mathcal{I}_{j-1})$. This is in done in [Osborne, 2010] for only two steps look ahead given the computational burden required to compute this loss for longer horizons. Note that analytical expression are only available in the myopic case $\Lambda_1(\mathbf{x}_*|\mathcal{I}_0)$.

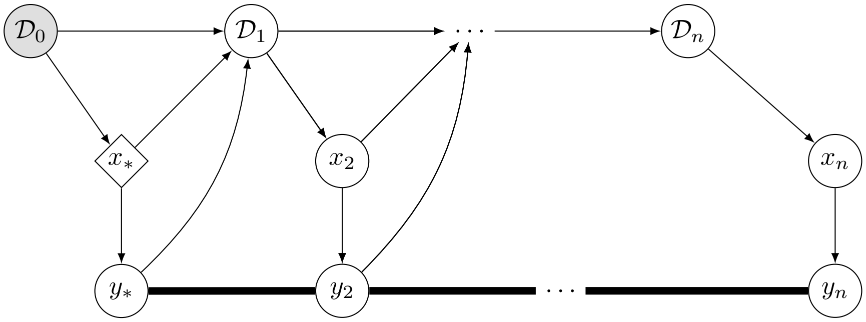


Figure 1: A Bayesian network describing the n-step lookahead problem. The shaded node ($\mathcal{D}_0$) is known, and the diamond node (x_*) is the current decision variable. All y nodes are correlated with one another under the GP model.

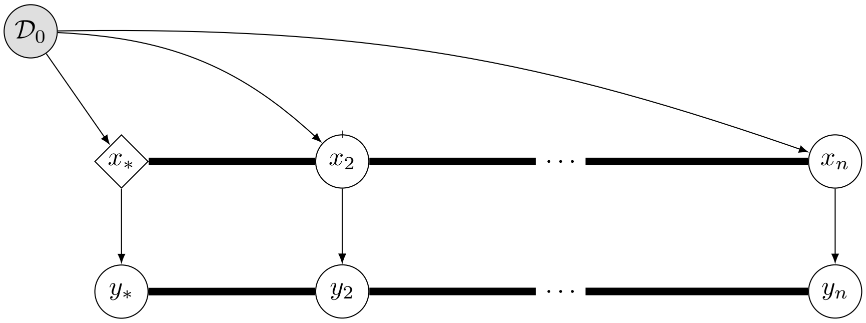


Figure 2: A Bayesian network describing our approximation to the n-step lookahead problem. The shaded node ($\mathcal{D}_0$) is known, and the diamond node (x_*) is the current decision variable, which is now directly connected with all future steps of the algorithm.

2.3 Contributions of this work

The goal of this work is *to propose an efficient approximation to Eq. (2) that will relieve the myopia of classical Bayesian optimization.* The precise contributions of this paper are:

- A new algorithm, GLASSES, to relieve the myopia of Bayesian optimisation that is able to efficiently take into account dozens of steps ahead. The method is based on the prediction of the future steps of the myopic algorithm to efficiently integrate out a long-side loss.

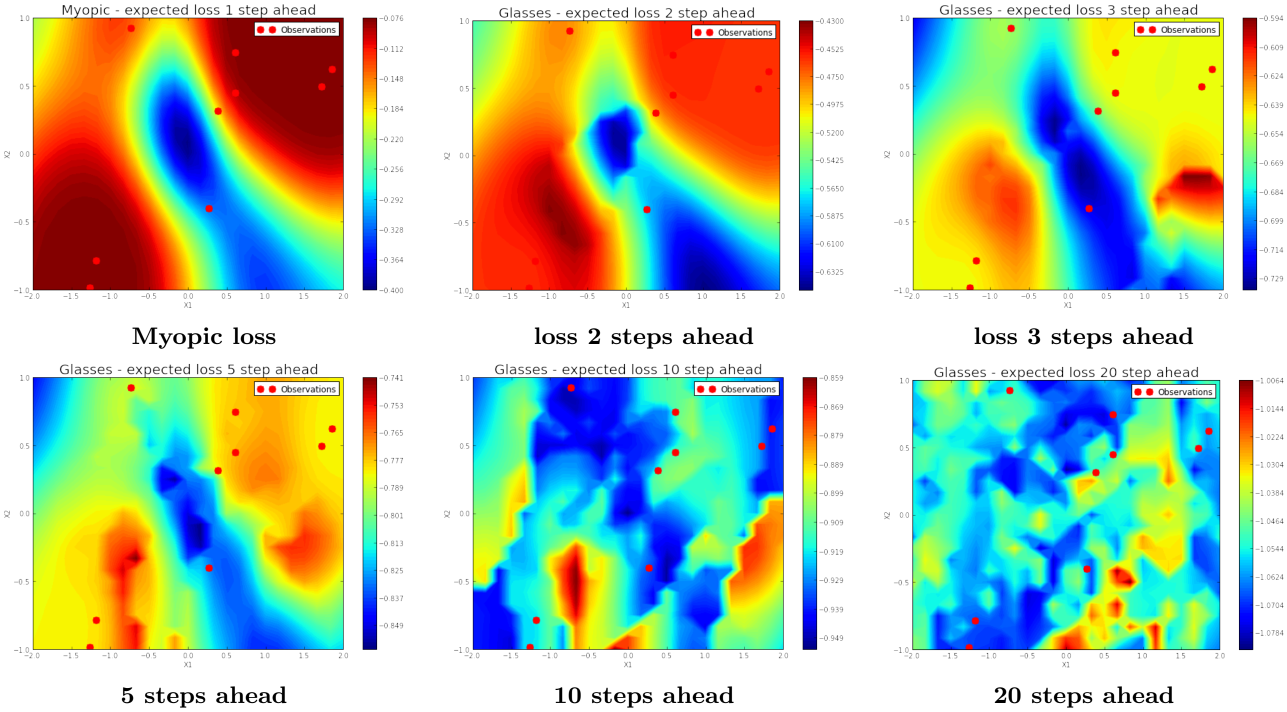
- The key aspect of our approach is to split the recursive optimization marginalization loop in Eq. (2) into two independent optimisation-marginalization steps that jointly act on all the future steps. We propose an Expectation-Propagation formulation for the joint marginalisation and we discuss different strategies to carry out the optimisation step.

- Together with this work, we deliver a *open source Python code framework* (link removed for blind review) containing a fully functional implementation of the method useful to reproduce the results of this work and applicable in general global optimisation problems. As we mentioned in the introduction of this work, there exist a limited literature in BO non-myopic methods and, to our knowledge, no BO package has any myopic loss functions among the available acquisition functions.

- Simulations: New practical experiments and insights that show that non-myopic methods outperform myopic approaches in a benchmark of optimisation problems.

3 The glasses Algorithm

As detailed in the previous section, a proper multi-step look ahead loss function requires the iterative optimization-marginalization of the future steps, which is computationally intractable. A possible way of dealing with this issue in to jointly modelling our epistemic uncertainty over the future locations $\mathbf{x}_2, \ldots, \mathbf{x}_n$ with a joint probability distribution



Myopic loss **loss 2 steps ahead** **loss 3 steps ahead**

5 steps ahead **10 steps ahead** **20 steps ahead**

Table 1: Estimated expected loss for different number of steps ahead in an example with 10 data points and the Six-hump Camel function. Increasing the steps ahead decreases global optimum of the loss; the algorithm will visit more future locations and therefore expected value of the best potential minimum decreases. Increasing the number of steps ahead flatten down the loss since it is likely for the algorithm to hit a good location irrespective of the initial point (all candidate points look better because of the future chances of the algorithm to be in a good location).

$p(\mathbf{x}_2, \ldots, \mathbf{x}_n | \mathcal{I}_0, \mathbf{x}_*)$ and to consider the expected loss

$$\Gamma_n(\mathbf{x}_*|\mathcal{I}_0) = \int \lambda(y_n) p(\mathbf{y}|\mathbf{X}, \mathcal{I}_0, \mathbf{x}_*) p(\mathbf{X}|\mathcal{I}_0, \mathbf{x}_*) d\mathbf{y} d\mathbf{x} \tag{3}$$

for $\mathbf{y} = \{y_*, \ldots, \ldots, \ldots, y_n\}$ the vector of future evaluations of f and $\mathbf{X}$ the $(n-1) \times d$ dimensional matrix whose rows represent the algorithm future locations. Note that $p(\mathbf{y}|\mathbf{X}, \mathcal{I}_0, \mathbf{x}_*)$ is multivariate Gaussian, since it corresponds to the predictive distribution of the GP at $\mathbf{X}$. This loss function corresponds the the graphical model in Figure 2. It differs from $\Lambda_n(\mathbf{x}_*|\mathcal{I}_0)$ in the fact that all future evaluations are modelled jointly rather then sequentially. This distribution, however, may be difficult to obtain. An natural option is to model $p(\mathbf{X}|\mathcal{I}_0, \mathbf{x}_*)$ using a continuous determinant point process (DPP) defined on $\mathcal{X}$ []. DPPs have nice computational properties in discrete sets. However, to integrate Eq. (3) by sampling from a DPP would require to do it conditioning to $\mathbf{x}_*$ and the number of steps ahead. Although this is possible in theory, the computational burden of this approach would be still very large.

An alternative and more efficient approach that we explore here is to work with a fixed set $\mathbf{X}$, which we assume it is given by some algorithm. As we show in this section, although this approach omits our epistemic uncertainty on X it drastically reduces the computational burden of approximating $\Lambda_n(\mathbf{x}_*|\mathcal{I}_0)$.

3.1 Oracle multiple steps look-ahead expected loss

Suppose that we had access to an oracle function $\mathcal{F}_n : \mathcal{X} \to \mathcal{X}^n$ able to predict the n future locations that the loss $\Lambda_n(\cdot)$ would suggest if we start evaluating f at $\mathbf{x}_*$. We assume that $\mathcal{F}_1(\mathbf{x}_*) = \mathbf{x}_*$, that is, the first visited location is always $\mathbf{x}_*$ itself. We work here under the assumption that the oracle has perfect information about the future locations, in the same way we have have perfect information about the locations that the algorithm already visited. This is obviously a totally unrealistic assumption in practice, but it will help us to set-up our algorithm. We leave for the next section the details of how to marginalise over the unknown $\mathcal{F}_n$.

Assume, for now, that $\mathcal{F}_n$ exists and that we have access to it we it and denote by $\mathbf{y} = \{y_*, \ldots, \ldots, \ldots, y_n\}$ the vector of future locations evaluations of f at $\mathcal{F}_n(\mathbf{x}_*)$.

Under this hypothesis it is possible to rewrite the expected loss in Eq. (2) as

$$\Lambda_n(\mathbf{x}_*|\mathcal{I}_0, \mathcal{F}_n(\mathbf{x}_*)) = \mathbb{E}[\min(\mathbf{y}, \eta)], \tag{4}$$

where the expectation is taken over the multivariate Gaussian distribution, with mean vector μ and covariance matrix Σ, that gives rise after marginalizing the posterior distribution of the GP at $\mathcal{F}_n(\mathbf{x}_*)$. See supplementary materials for details.

The intuition behind Eq. (4) is as follows: the expected loss at $\mathbf{x}_*$ is the best possible function value that we expect to find in the next n steps, conditional on the first evaluation being made at $\mathbf{x}_*$. The expected loss depends not just on the next function evaluation, but how we expect to benefit from the remaining $n-1$ evaluations. See Figure 2

To compute Eq. (4) we propose to use Expectation Propagation (EP) [Minka, 2001]. This turns out to be a natural operation by observing that

$$\begin{aligned}\mathbb{E}[\min(\mathbf{y}, \eta)] &= \eta \int_{\mathbb{R}^n} \prod_{i=1}^{n} h_i(\mathbf{y}) \mathcal{N}(\mathbf{y}; \mu, \Sigma) \mathrm{d}\mathbf{y} \qquad (5)\\ &+ \sum_{j=1}^{n} \int_{\mathbb{R}^n} y_j \prod_{i=1}^{n} t_{j,i}(\mathbf{y}) \mathcal{N}(\mathbf{y}; \mu, \Sigma) \mathrm{d}\mathbf{y}\end{aligned}$$

where $h_i(\mathbf{y}) = \mathbb{I}\{y_i > \eta\}$ and

$$t_{j,i}(\mathbf{y}) = \begin{cases} \mathbb{I}\{y_j \leq \eta\} & \text{if i=j} \\ \mathbb{I}\{0 \leq y_i - y_j\} & \text{otherwise.} \end{cases}$$

See supplementary materials for details. The first term in Eq. (5) is a Gaussian probability on unbounded polyhedron in which the limits are aligned with the axis. The second term is the sum of the Gaussian expectations on different non-axis-aligned different polyhedra defined by the indicator functions. Both terms can be computed with EP using the approach proposed in [Cunningham et al., 2011]. In a nutshell, to compute the integrals one need to replace the indicator functions with univariate Gaussian that play the role of *soft-indicators* in the EP iterations. This method is computationally efficient and scales well for high dimensions. Note that when $n = 1$, (4) reduces to (1).

Under the hypothesis of this section, the next evaluation is located where $\Lambda_n(\mathbf{x}_*|\mathcal{I}_0, \mathcal{F}_n(\mathbf{x}_*))$ gives the minimum value.

3.2 Predicting the future steps of BO

- The goal of this section is to define some $\hat{\mathcal{F}}(\mathbf{x}_*)$.
- One could take the MAP of the dpp! expensive.
- One can use a batch bo method.
- We use one that is fast, and imitates the optimization-marginalization by a optimizaiton-penalization.
- Describe briefly
- Example
- Repulsion effect, connection with dpp

Name	Function domain	D
Cosines	$[0, 1] \times [0, 1]$	2
Branin	$[-5, 10] \times [-5, 10]$	2
Sixhumpcamel	$[-2, 2] \times [-1, 1]$	2
McCormick	$[-1.5, 4] \times [-3, 4]$	2
Goldstein	$[-2, 2] \times [-2, 2]$	2
Egg-holder	$[-512, 512] \times [-512, 512]$	2
Powers	$[-1, 1] \times [-1, 1]$	2
Alpine2-2	$[-10, 10]^D$	2
Alpine2-5		5
Alpine2-10		10
gSobol-2	$[-5, 5]^D$	2
gSobol-5		5
gSobol-10		10

Table 3: Details of the functions used in the experiments

$$[\mathcal{F}_n(\mathbf{x}_*)]_k = \arg\max_{x \in \mathbf{x}} \left\{ g(\Lambda_1(\mathbf{x}_*|\mathcal{I}_0)) \prod_{j=1}^{k-1} \varphi(\mathbf{x}; \hat{\mathbf{x}}_j) \right\}, \tag{6}$$

where $\varphi(\mathbf{x}; \mathbf{x}_j)$ are local local penalizers centered at $\mathbf{x}_j$ and $g : \mathbb{R} \to \mathbb{R}^+$ the *soft-plus* transformation $g(z) = \ln(1 + e^z)$ elsewhere.

3.3 Algorithm and computational details

4 Results

4.1 Testing the validity of the approach

To study the validity of our approximation we choose a variety of functions with a range of dimensions and domains domain sizes. See Table **??** for details.

5 Conclusions

References

Javad Azimi, Ali Jalali, and Xiaoli Fern. Dynamic batch Bayesian optimization. *CoRR*, abs/1110.3347, 2011.

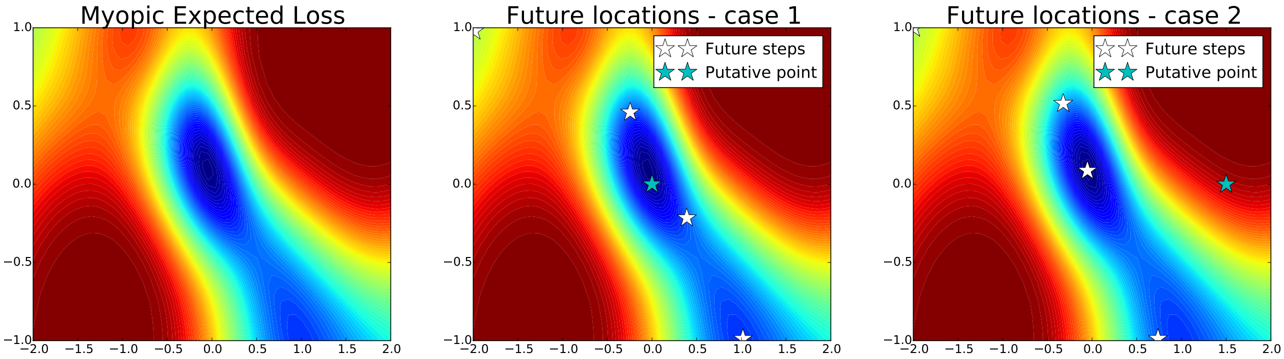


Table 2:

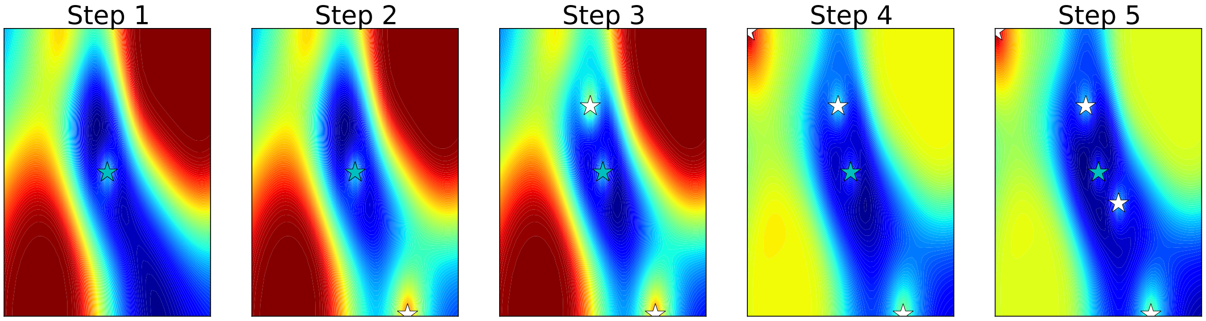


Figure 3: ffff

Algorithm 1 Decision process of the GLASSES algorithm.

```
Input: dataset $\mathcal{D}_0 = \{(\mathbf{x}_0, y_0)\}$, number of remaining evaluations $(n)$, look-ahead predictor $\mathcal{F}$.
for $j = 0$ to $n$ do
    1. Fit a GP with kernel $k$ to $\mathcal{D}_j$.
    2. Build a predictor of the future $n - l$ evaluations: $\mathcal{F}_{n-j}(\mathbf{x}_*)$.
    3. Select next location $\mathbf{x}_j$ by taking $\mathbf{x}_j = \arg\min_{\mathbf{x} \in \mathcal{X}} \Lambda_{n-j}(\mathbf{x}_* | \mathcal{I}_0, \mathcal{F}_{n-j}(\mathbf{x}_*))$.
    4. Evaluate $f$ at $\mathbf{x}_j$ and obtain $y_j$.
    5. Augment the dataset $\mathcal{D}_j = \{\mathcal{D}_{j-1} \cup (\mathbf{x}_j, y_j)\}$.
end for
Returns: New location at $\arg\min_{x \in \mathcal{X}} \{\mu_2(\mathbf{x})\}$.
```

Javad Azimi, Ali Jalali, and Xiaoli Zhang Fern. Hybrid batch Bayesian optimization. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Eric Brochu, Vlad M. Cora, and Nando 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.

John P. Cunningham, Philipp Hennig, and Simon Lacoste-Julien. Gaussian probabilities and expectation propagation. *arXiv:1111.6832 [stat]*, Nov 2011. arXiv: 1111.6832.

R. Garnett, M. A. Osborne, and S. J. Roberts. *Bayesian optimization for sensor set selection*, page 209–219. ACM, 2010. ISBN 1605589888. doi: 10.1145/1791212.1791238.

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. A multi-points criterion for deterministic parallel global optimization based on Gaussian processes. *HAL: hal-00260579*, 2009.

Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

Daniel James Lizotte. *Practical Bayesian Optimization.* PhD thesis, University of Alberta, 2008. AAINR46365.

Roman Marchant, Fabio Ramos, and Scott Sanner. Sequential Bayesian optimisation for spatial-temporal

noise-free	EL	GL-2	GL-3	GL-5	GL-10	GL-(n-k)
Cosines	-1.562084	-1.577403	-1.586372	**-1.600443**	-1.465383	
Branin	**0.727696**	1.018735	1.865223	1.847867	1.190992	
Sixhumpcamel	**-1.111226**	-1.099144	-1.085236	-1.091393	-1.104145	
McCormick	-1.939604	-1.936456	-1.916778	-1.950658	**-2.003377**	
Goldstein						
Egg-holder						
Powers	-0.125614	**-0.185978**	-0.149429	-0.158541	-0.180303	
Alpine2 (d=2)						
Alpine2 (d=5)						
Alpine2 (d=10)						
gSobol (d=2)						
gSobol (d=5)						
gSobol (d=10)						

$sd. = 0.1$	EL	GL-2	GL-3	GL-5	GL-10	GL-(n-k)
Cosines	-1.562084	-1.577403	-1.586372	**-1.600443**	-1.465383	
Branin	**0.727696**	1.018735	1.865223	1.847867	1.190992	
Sixhumpcamel	**-1.111226**	-1.099144	-1.085236	-1.091393	-1.104145	
McCormick	-1.939604	-1.936456	-1.916778	-1.950658	**-2.003377**	
Powers	-0.125614	**-0.185978**	-0.149429	-0.158541	-0.180303	
Goldstein						
Egg-holder						
Alpine2 (d=2)						
Alpine2 (d=5)						
Alpine2 (d=10)						
gSobol (d=2)						
gSobol (d=5)						
gSobol (d=10)						

Table 4: Results for the mean of the replicates

monitoring. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2014.

Ruben Martinez-Cantin, Nando de Freitas, Eric Brochu, José Castellanos, and Arnaud Doucet. A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27(2): 93–103, August 2009. ISSN 0929-5593, 1573-7527. doi: 10.1007/s10514-009-9130-2.

Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 362–369, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-800-1.

Michael Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature.* PhD thesis, PhD thesis, University of Oxford, 2010.

Michael A. Osborne, Roman Garnett, and Stephen J. Roberts. Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization (LION3)*, pages 1–15, 2009.

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

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. *Practical Bayesian optimization of machine learning algorithms*, page 2951–2959. 2012.

Simon Streltsov and Pirooz Vakili. A non-myopic utility function for statistical global optimization algorithms. *J. Global Optim.*, 14(3):283–298, 1999.

Supplementary materials for: ‘GLASSES: Relieving The Myopia Of Bayesian Optimisation”

Authors here

S1 Oracle Multiple Steps loook-ahead Expected Loss

Denote by $\eta_n = \min\{\mathbf{Y}_0, y_*, y_2 \dots, y_{n-1}\}$ the value of the best visited location when looking at n evaluations in the future. Note that η_n reduces to the current best lost η in the one step-ahead case. It is straightforward to see that

$$\min(y_n, \eta_n) = \min(\mathbf{y}, \eta).$$

It holds hat

$$\Lambda_n(\mathbf{x}_*|\mathcal{I}_0, \mathcal{F}_n(\mathbf{x}_*)) \quad = \quad \int \min(\mathbf{y}, \eta) \prod_{j=1}^{n} p(y_j|\mathcal{I}_{j-1}, \mathcal{F}_n(\mathbf{x}_*))\mathrm{d}y_* \dots \mathrm{d}y_n$$

where the integrals with respect to $\mathbf{x}_2 \dots \mathrm{d}\mathbf{x}_n$ are $p(\mathbf{x}_j|\mathcal{I}_{j-1}, \mathcal{F}_n(\mathbf{x}_*)) = 1$, $j = 2, \dots, n$ since we don’t need to optimize for any location and $p(y_j|\mathbf{x}_j, \mathcal{I}_{j-1}, \mathcal{F}_n(\mathbf{x}_*)) = p(y_j|\mathcal{I}_{j-1}, \mathcal{F}_n(\mathbf{x}_*))$. Notice that

$$\begin{aligned}
\prod_{j=1}^{n} p(y_j|\mathcal{I}_{j-1}, \mathcal{F}_n(\mathbf{x}_*)) \quad &= \quad p(y_n|\mathcal{I}_{n-1}, \mathcal{F}_n(\mathbf{x}_*)) \prod_{j=1}^{n-1} p(y_j|\mathcal{I}_{j-1}\mathcal{F}_n(\mathbf{x}_*)) \\
&= \quad p(y_n, y_{n-1}|\mathcal{I}_{n-2}, \mathcal{F}_n(\mathbf{x}_*)) \prod_{j=1}^{n-2} p(y_j|\mathcal{I}_{j-1}\mathcal{F}_n(\mathbf{x}_*)) \\
& \quad\;\;\; \dots \\
&= \quad p(y_n, y_{n-1}, \dots, y_2|\mathcal{I}_1, \mathcal{F}_n(\mathbf{x}_*)) \prod_{j=1}^{2} p(y_j|\mathcal{I}_{j-1}\mathcal{F}_n(\mathbf{x}_*)) \\
&= \quad p(\mathbf{y}|\mathcal{I}_0, \mathcal{F}_n(\mathbf{x}_*))
\end{aligned}$$

and therefore

$$\Lambda_n(\mathbf{x}_*|\mathcal{I}_0, \mathcal{F}_n(\mathbf{x}_*)) = \mathbb{E}[\min(\mathbf{y}, \eta)] = \int \min(\mathbf{y}, \eta) p(\mathbf{y}|\mathcal{I}_0, \mathcal{F}_n(\mathbf{x}_*)) d\mathbf{y}$$

S2 Formulation of the Oracle Multiple Steps loook-ahead Expected Loss to be computed using Expectation Propagation

Assume that $\mathbf{y} \sim \mathcal{N}(\mathbf{y}; \mu, \Sigma)$. Then we have that

$$\begin{aligned}
\mathbb{E}[\min(\mathbf{y}, \eta)] \quad &= \quad \int_{\mathbb{R}^n} \min(\mathbf{y}, \eta) \mathcal{N}(\mathbf{y}; \mu, \Sigma) d\mathbf{y} \\
&= \quad \int_{\mathbb{R}^n - (\eta, \infty)^n} \min(\mathbf{y}) \mathcal{N}(\mathbf{y}; \mu, \Sigma) d\mathbf{y} + \int_{(\eta, \infty)^n} \eta \mathcal{N}(\mathbf{y}; \mu, \Sigma) d\mathbf{y}.
\end{aligned}$$

The first term can be written as follows:

$$\int_{\mathbb{R}^n - (\eta, \infty)^n} \min(\mathbf{y}) \mathcal{N}(\mathbf{y}; \mu, \Sigma) d\mathbf{y} = \sum_{j=1}^{n} \int_{P_j} y_j \mathcal{N}(\mathbf{y}; \mu, \Sigma) \mathrm{d}\mathbf{y}$$

where $P_j := \{\mathbf{y} \in \mathbb{R}^n - (\eta, \infty)^n : y_j \leq y_i, \ \forall i \neq j\}$. We can do this because the regions P_j are disjoint and it holds that $\cup_{j=1}^n P_j = \mathbb{R}^n - (\eta, \infty)^n$. Also, note that the $\min(\mathbf{y})$ can be replaced within the integrals since within each P_j it holds that $\min(\mathbf{y}) = y_j$. Rewriting the integral in terms of indicator functions we have that

$$\sum_{j=1}^n \int_{P_j} y_j \mathcal{N}(\mathbf{y}; \mu, \Sigma) \mathrm{d}\mathbf{y} = \sum_{j=1}^n \int_{\mathbb{R}^n} y_j \prod_{i=1}^n t_{j,i}(\mathbf{y}) \mathcal{N}(\mathbf{y}; \mu, \Sigma) \mathrm{d}\mathbf{y} \tag{S.1}$$

where $t_{j,i}(y) = \mathbb{I}\{y_i \leq \eta\}$ if $j = i$ and $t_{j,i}(y) = \mathbb{I}\{y_j \leq y_i\}$ otherwise.

The second term can be written as

$$\int_{(\eta,\infty)^n} \eta \mathcal{N}(\mathbf{y}; \mu, \Sigma) d\mathbf{y} = \eta \int_{\mathbb{R}^n} \prod_{i=1}^n h_i(\mathbf{y}) \mathcal{N}(\mathbf{y}; \mu, \Sigma) d\mathbf{y} \tag{S.1}$$

where $h_i(\mathbf{y}) = \mathbb{I}\{y_i > \eta\}$. Merge (S.1) and (S2) to obtain Eq. (5).