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 [15], sensor selection [5] or experimental design [10]. 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. [13] 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. [9] 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.

There is a clear link between the multi-step lookahead problem and that considered in the literature as *batch* Bayesian optimisation. The two problems are distinct but related: the multi-step lookahead problem requires the challenging marginalisation over unknown future evaluation *locations*, in addition to the unknown future evaluation *values* also marginalised by batch approaches. Similarly to the state-of-the-art in multi-step lookahead, the batch literature provides only poor scaling with the number of evaluations. [6] present results for no more than six simultaneous function evaluations. [1, 2] use the surrogate model for f to generate 'fake' observations and avoid the marginalization step. This produce a large accumulation of errors that does not allow the use of these techniques for the collection of large batches.

We propose an algorithm, GLASSES, that provides scaling superior to existing alternatives.

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

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

$\mathbf{x}_M = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$. 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 [8, 7, 15, 3].

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 $(\mathbf{K}_\theta)_{ij} = k_\theta(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}_\theta(\mathbf{x}_*) = [k_\theta(\mathbf{x}_1, \mathbf{x}_*), \dots, k_\theta(\mathbf{x}_n, \mathbf{x}_*)]^\top$ [14].

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

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

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 [12].

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 [12] as

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

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 2.

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 [12] 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 $\Lambda_1(\mathbf{x}_*|\mathcal{I}_0)$.

2.3 contribution of this work

The goal of this work is *to propose a scalable approximation to Eq. (2) that will relieve the myopia of classical Bayesian optimization methods.* The key aspect of our approach is to split the recursive optimization marginalization loop in Eq. (2) into two independent optimization-marginalization steps that jointly act on all the future steps. The precise contributions of this paper are:

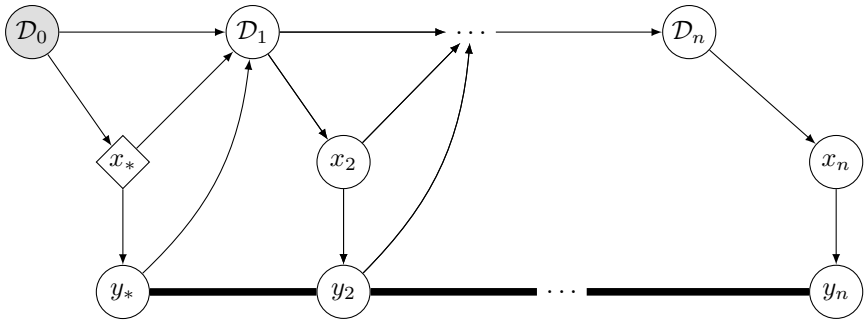


Figure 1: A Bayesian network describing the n-step lookahead problem. The shaded node ($\mathcal{D}_0$) is known, and the diamond node (x_*) is the current decision variable. All y nodes are correlated with one another under the GP model.

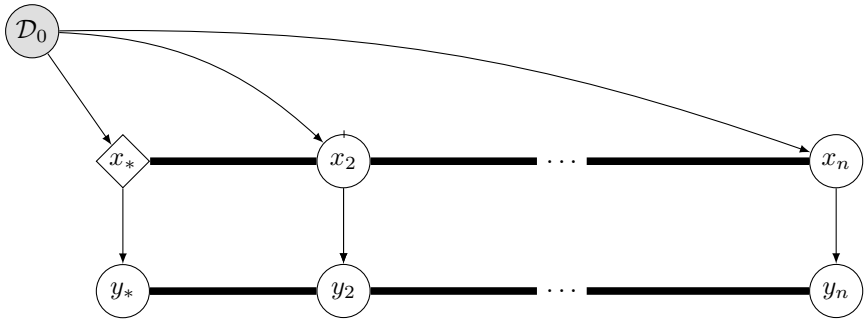


Figure 2: A Bayesian network describing our approximation to the n-step lookahead problem.

- Method, it is cool!, etc
- Code Framework DESCRIBE - GPyOpt, etc
- Any theoretical property (can we do something here?)
- Simulations: New pratical insights in the direction that non-myopic methods are the way.Nice bit! perhaps the most important of the paper from a practical perspective.
- Question here: How would one define a non-myopic version of Entropy Search?? just curious... MO says: it'd be possible, but even more crazily expensive. You'd need to see what the entropy in the location of the minimum is after the next n evaluations, and then marginalise over the locations and values of the last n-1 of them.

3 The GLASSES Algorithm

3.1 Oracle multiple steps loook-ahead expected loss

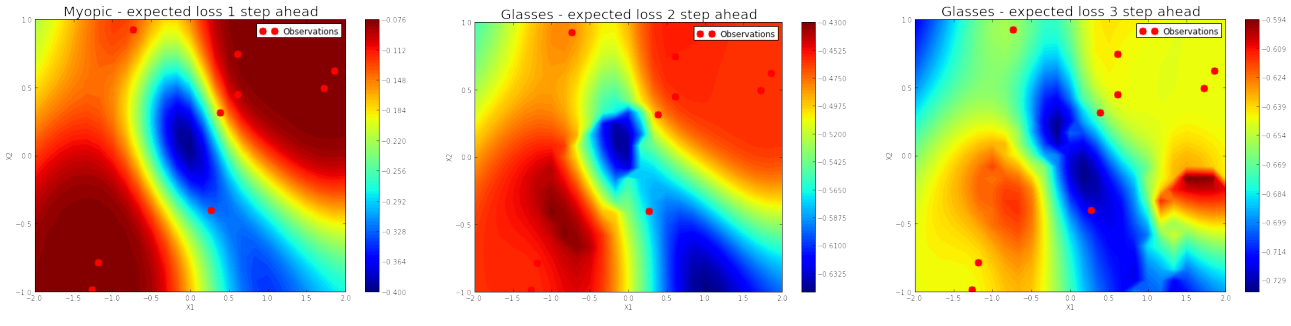
Suppose that we had access to an oracle function $\mathcal{F}_n : \mathcal{X} \to \mathcal{X}^n$ able to predict the n future locations that the loss $\Lambda_n(\cdot)$ would suggest if we start evaluating f at $\mathbf{x}_*$. We assume that $\mathcal{F}_1(\mathbf{x}_*) = \mathbf{x}_*$, that is, the 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 $\mathbf{y} = \{y_*, \dots, \dots, y_n\}$ 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

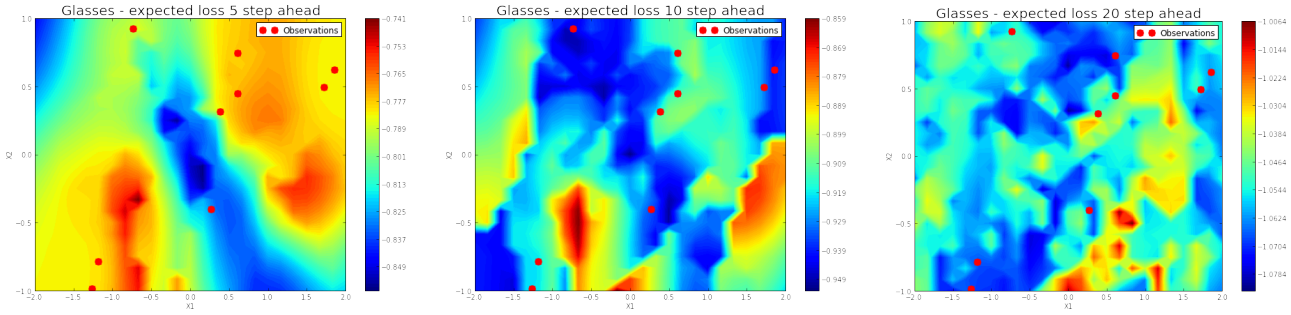
$$\Lambda_n(\mathbf{x}_*|\mathcal{I}_0, \mathcal{F}_n(\mathbf{x}_*)) = \mathbb{E}[\min(\mathbf{y}, \eta)], \tag{3}$$

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. (S.1) 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,



Myopic loss | 2 steps ahead | 3 steps ahead



5 steps ahead' | 10 steps ahead | 20 steps ahead

Table 1: (PRELIMINAR PLOTS) Estimated expected loss for differnt number of steps ahead.

but how we expect to benefit from the remaining $n-1$ evaluations.

NOT CLEAR TO ME YET HOW TO FINISH THIS. THIS IS PROBABLY THE MOST IMPORTANT SENTENCE OF THE PAPER... INCLUDE AN FIGURE TO SUPPORT IT? MO: a figure is an excellent idea!

To compute Eq. (S.1) we propose to use Expectation Propagation (EP) [11]. This turns out to be a natural operation by observing that

$$\begin{aligned}\mathbb{E}[\min(\mathbf{y},\eta)] &= \eta \int_{\mathbb{R}^n} \prod_{i=1}^{n} h_i(\mathbf{y}) \mathcal{N}(\mathbf{y};\mu,\Sigma)\mathrm{d}\mathbf{y} \qquad (4)\\ &+ \sum_{j=1}^{n} \int_{\mathbb{R}^n} y_j \prod_{i=1}^{n} t_{j,i}(\mathbf{y}) \mathcal{N}(\mathbf{y};\mu,\Sigma)\mathrm{d}\mathbf{y}\end{aligned}$$

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. (4) 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 easily computed with EP using the approach proposed in [4]. 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 $n = 1$, (S.1) 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.

NOTE HERE ON GRADIENTS OF $\Lambda_n(\mathbf{x}_*|\mathcal{I}_0, \mathcal{F}_n(\mathbf{x}_*))$: should be possible to derive but, in practice EP will provide and approximation to the integral so it may not be the best thing to do. Using DIRECT sounds to me here like the best option.

Good point, and I think using DIRECT should be fine. That said, if you did want gradients, what about autograd: https://github.com/HIPS/autograd/?

3.2 Back to the Future: Predicting the future steps of BO

3.3 Algorithm and computational details

Algorithm 1 Decision process of the GLASSES algorithm.

Input: dataset $\mathcal{D}_0 = \{(\mathbf{x}_0, y_0)\}$, number of remaining evaluations (n).
Fit a GP with kernel k to $\mathcal{D}_0$.
Select $\mathbf{x}_{1*}, \ldots, \mathbf{x}_{r*}$ representer points of the loss.
for $j = 1$ **to** r **do**
  Take s samples from a conditional n-DPP of kernel k given $\mathbf{x}_{j*}$.
  Approximate the expected loss at $\mathbf{x}_j^*$ for the s samples computing $E[\min(\mathbf{y}, \eta)]$.
  Average the expected loss for the s samples and obtain $\tilde{\Lambda}_n(\mathbf{x}_j^*)$.
end for
Approximate $\Lambda_n(\mathbf{x}_*)$ fitting a GP_2 to $\{(\mathbf{x}_{j*}, \tilde{\Lambda}_n(\mathbf{x}_{j*})\}_{j=1}^r$ with posterior mean μ_2.
Returns: New location at $\arg\min_{x \in \mathcal{X}} \{\mu_2(\mathbf{x})\}$.

problem	EL	GL-2	GL-3	GL-5	GL-10
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

Table 2: Results for the mean of the replicates

4 Results

5 Conclusions

References

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

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

[3] 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.

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

[5] R. Garnett, M. A. Osborne, and S. J. Roberts. *Bayesian optimization for sensor set selection*, page 209–219. ACM, 2010.

[6] 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.

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

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

[9] Roman Marchant, Fabio Ramos, and Scott Sanner. Sequential Bayesian optimisation for spatial-temporal monitoring. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2014.

[10] 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.

[11] 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.

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

[13] 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.

problem	EL	GL-2	GL-3	GL-5	GL-10
cosines	-1.641266	-1.614421	-1.647324	**-1.685686**	-1.496753
branin	**0.501499**	0.776298	1.223222	0.791490	0.626915
sixhumpcamel	**-1.115223**	-1.096584	-1.073318	-1.098454	-1.103938
mccormick	-1.970486	-1.996537	-1.966840	-1.998333	**-2.031038**
powers	-0.123350	-0.159824	-0.152354	-0.125650	**-0.172454**

Table 3: Results for the median of the replicates

problem	EL	GL-2	GL-3	GL-5	GL-10
cosines	-1.700992	-1.899286	-1.803843	-1.815387	**-1.935917**
branin	0.395783	**0.282889**	0.367038	0.305713	0.388108
sixhumpcamel	-1.225345	-1.240747	-1.170774	-1.201114	**-1.296890**
mccormick	**-2.128225**	-2.126073	-2.038700	-2.055372	-2.077451
powers	-0.184954	**-0.350845**	-0.267484	-0.287553	-0.252537

Table 4: Results for the median of the replicates

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

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

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

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

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

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

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

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

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

$$\sum_{j=1}^{n} \int_{P_j} y_j \mathcal{N}(\mathbf{y}; \mu, \Sigma) \mathrm{d}\mathbf{y} = \sum_{j=1}^{n} \int_{\mathbb{R}^n} y_j \prod_{i=1}^{n} t_{j,i}(\mathbf{y}) \mathcal{N}(\mathbf{y}; \mu, \Sigma) \mathrm{d}\mathbf{y} \tag{S.2}$$

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

$$\int_{(\eta,\infty)^n} \eta \mathcal{N}(\mathbf{y}; \mu, \Sigma) d\mathbf{y} = \eta \int_{\mathbb{R}^n} \prod_{i=1}^{n} h_i(\mathbf{y}) \mathcal{N}(\mathbf{y}; \mu, \Sigma) d\mathbf{y} \tag{S.2}$$

where $h_i(\mathbf{y}) = \mathbb{I}\{y_i > \eta\}$. Merge (S.2) and (S2) to obtain Eq. (4).