BINOCULARS for Efficient, Nonmyopic Sequential Experimental Design: Supplementary Material

Shali Jiang [*1] **Henry Chai** [*1] **Javier Gonzalez** [2] **Roman Garnett** [1]

A. Decomposing the Expected Batch Utility

The expected marginal utility of designing the batch of experiments X is

$$Q(X \mid \mathcal{D}) = \mathbb{E}_Y[u(Y \mid X, \mathcal{D})], \tag{1}$$

where the expectation is taken over the joint distribution of $Y = \{y_1, \ldots, y_T\}$, $p(Y \mid X, \mathcal{D})$.

Let x_j be an arbitrary point in the batch X and let $X_{-j} = X \setminus \{x_j\}$. Through the use of a telescoping sum trick, $Q(X \mid \mathcal{D})$ can be decomposed as follows:

$$\begin{aligned} Q(X \mid \mathcal{D}) &= \mathbb{E}_Y\Big[u\big(\mathcal{D} \cup \{(X, Y)\}\big) - u\big(\mathcal{D}\big)\Big] \\ &= \mathbb{E}_Y\Big[u\big(\mathcal{D} \cup \{(X, Y)\}\big) - u\big(\mathcal{D} \cup \{(x_j, y_j)\}\big) \\ &\qquad + u\big(\mathcal{D} \cup \{(x_j, y_j)\}\big) - u\big(\mathcal{D}\big)\Big]. \end{aligned} \tag{2}$$

Rewriting the set in the first term as

$$\mathcal{D} \cup \{(X, Y)\} = \big(\mathcal{D} \cup \{(x_j, y_j)\}\big) \cup \{(X_{-j}, Y_{-j})\}, \tag{3}$$

the derivation in (2) can be continued as follows:

$$\begin{aligned} &Q(X \mid \mathcal{D}) \\ &\quad = \mathbb{E}_Y\Big[u\big(\mathcal{D} \cup \{(X, Y)\}\big) - u\big(\mathcal{D} \cup \{(x_j, y_j)\}\big) \\ &\qquad\quad + u\big(\mathcal{D} \cup \{(x_j, y_j)\}\big) - u\big(\mathcal{D}\big)\Big] \\ &\quad = \mathbb{E}_Y\Big[u\big(Y_{-j} \mid X_{-j}, \mathcal{D} \cup \{(x_j, y_j)\}\big) + u\big(y_j \mid x_j, \mathcal{D}\big)\Big]. \end{aligned} \tag{4}$$

Finally, to achieve the desired result, the expectation w.r.t. $p(Y \mid X, \mathcal{D})$ can be rewritten as nested expectations w.r.t. $p(y_j \mid x_j, \mathcal{D})$ and $p(Y_{-j} \mid X_{-j}, \mathcal{D} \cup \{(x_j, y_j)\})$:

$$\begin{aligned} Q(X \mid \mathcal{D}) &= \mathbb{E}_{y_j}\Big[\mathbb{E}_{Y_{-j}}\Big[u\big(Y_{-j} \mid X_{-j}, \mathcal{D} \cup \{(x_j, y_j)\}\big) \\ &\qquad + u\big(y_j \mid x_j, \mathcal{D}\big)\Big]\Big] \\ &= \mathbb{E}_{y_j}\Big[u\big(y_j \mid x_j, \mathcal{D}\big)\Big] + \mathbb{E}_{y_j}\Big[Q\big(X_{-j} \mid \mathcal{D} \cup \{(x_j, y_j)\}\big)\Big]. \end{aligned} \tag{5}$$

B. Full-Lookahead Expected Improvement as Bellman Equation

When $T = 1$, i.e., there is only one evaluation left, the optimal policy degenerates to the simplest case known as *expected improvement* (EI):

$$x^* = \arg\max_x EI_1(x) \equiv \mathbb{E}[(f(x) - y_0)^+]. \tag{6}$$

Now consider $T = 2$. Starting from location x, the improvement of the next two evaluations depends on three random variables: $y \equiv f(x)$, the next evaluation location x', and its value $y' \equiv f(x')$; computing the expected utility of starting from x requires integrating all three variables out:

$$\begin{aligned} EI_2(x) = \int_{y, x', y'} &(\max\{y_1, y_2\} - y_0)^+ \, p(y \mid x) \\ &p(x' \mid x, y) p(y' \mid x', y, x) dy dx' dy'. \end{aligned} \tag{7}$$

Given

$$(\max\{y, y'\} - y_0)^+ = (y - y_0)^+ + (y' - \max(y_0, y))^+$$

(Ginsbourger et al., 2010), we have

$$\begin{aligned} EI_2(x) &= \int_y (y - y_0)^+ dy \\ &\quad + \int_y \int_{x'} \int_{y'} (y' - \max(y_0, y))^+ \, p(y' \mid x', y, x) dy' \\ &\qquad p(x' \mid x, y) dx' p(y \mid x) dy \\ &= EI_1(x) \\ &\quad + \int_y \int_{x'} EI_1(x' \mid x, y) p(x' \mid x, y) dx' p(y \mid x) dy \end{aligned} \tag{8}$$

[*]Equal contribution [1]Department of Computer Science and Engineering, Washington University in Saint Louis, Saint Louis, Missouri, USA [2]Microsoft Research Cambridge, Cambridge, UK. Correspondence to: Shali Jiang <jiang.s@wustl.edu>, Henry Chai <hchai@wustl.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

By Bellman's principle of optimality, we have

$$p(x' \mid x, y) = \delta(x' - \arg\max_{x^*} EI_1(x^* \mid x, y)). \tag{9}$$

Therefore,

$$\int_{x'} EI_1(x' \mid x, y) p(x' \mid x, y) dx' = \max_{x'} EI_1(x' \mid x, y), \tag{10}$$

and hence

$$EI_2(x) = EI_1(x) + \mathbb{E}[\max_{x'} EI_1(x' \mid x, y)]. \tag{11}$$

In general, we have the following Bellman equation for k-step expected utility

$$EI_k(x) = EI_1(x) + \mathbb{E}[\max_{x'} EI_{k-1}(x' \mid x, y)]. \tag{12}$$

C. Additional Bayesian Optimization Results

In the main paper, we presented BO results for nine synthetic functions. These nine functions are selected from the 31 functions shown in Table 1, with gap of EI less than 0.9. We only run up to 10.EI for all functions, so 12.EI.s and 15.EI.s are not shown. We argue that by identifying this set of "hard" functions, we are able to consistently see the advantage of nonmyopic BO methods. In Table 1, we can see all variants of our method perform better than EI on average, but other interesting patterns are weak, possibly because they are averaged out by the "easy" functions.

Table 2 includes results of rollout and GLASSES on five synthetic functions, after removing four from the nine for which the optima are located in the center of the domain. We remove these functions because the DIRECT optimization procedure used in our implementations of rollout and GLASSES always starts evaluating exactly at the center of the domain. Thus the performance of these methods on benchmarks where the global optimum just happens to be in the center is artificially inflated. This artifact was also pointed out in Lam et al. (2016); Wu and Frazier (2019) also excluded such results because of this.

We surprisingly see rollout and GLASSES perform even worse than EI on average for these five functions. This is an indicator that the synthetic benchmark functions are very different than the real-world functions. Note that Malkomes and Garnett (2018) also observed significantly different results on synthetic and real functions in their unrelated BO experiments.

Table 3 shows the average results of 50 repeats of EI and both "sampling" and "best" variants of q.EI on the real world functions. Different from the results on synthetic functions, we do not see "sampling" being consistently better than "best" or the other way around.

References

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, pages 131–162. 2010.

Remi Lam, Karen Willcox, and David H. Wolpert. Bayesian optimization with a finite budget: an approximate dynamic programming approach. In *Advances in Neural Information Processing Systems (*NEURIPS*) 29*, 2016.

Gustavo Malkomes and Roman Garnett. Automating Bayesian optimization with Bayesian optimization. In *Advances in Neural Information Processing Systems (*NEURIPS*) 31*, 2018.

Jian Wu and Peter Frazier. Practical two-step lookahead Bayesian optimization. In *Advances in Neural Information Processing Systems (*NEURIPS*) 32*, 2019.

Table 1: Average gap of 30 repeats on all 31 synthetic functions.

	EI	2.EI.b	2.EI.s	3.EI.b	3.EI.s	4.EI.b	4.EI.s	5.EI.b	5.EI.s	10.EI.b	10.EI.s
branin	*1.000*	1.000	0.999	1.000	0.999	*1.000*	1.000	*1.000*	1.000	**1.000**	0.999
rosenbrock2	*0.989*	*0.978*	*0.985*	*0.990*	*0.981*	0.971	*0.979*	*0.969*	**0.996**	*0.981*	0.973
rosenbrock4	*0.989*	*0.989*	0.988	0.990	*0.990*	*0.991*	*0.990*	**0.992**	0.988	*0.991*	*0.989*
rosenbrock6	*0.989*	0.989	*0.990*	**0.992**	*0.990*	*0.990*	*0.990*	*0.991*	*0.990*	*0.991*	0.985
hartmann3	1.000	*1.000*	*1.000*	1.000	*1.000*	**1.000**	*1.000*	*1.000*	*1.000*	*0.999*	*1.000*
hartmann6	0.957	0.966	0.964	0.970	0.965	0.974	0.970	*0.976*	0.974	**0.978**	0.971
eggholder	*0.605*	*0.606*	*0.589*	*0.603*	*0.612*	*0.649*	*0.638*	0.554	*0.620*	*0.600*	**0.651**
dropwave	0.455	0.489	0.524	0.475	*0.599*	*0.538*	0.550	0.435	*0.613*	0.448	**0.651**
beale	*0.920*	*0.903*	*0.910*	**0.935**	*0.915*	*0.927*	0.874	*0.901*	0.902	*0.912*	0.900
shubert	0.323	0.299	*0.440*	*0.387*	**0.551**	0.382	*0.500*	*0.464*	0.371	0.285	*0.458*
sixhumpcamel6	*0.996*	0.994	0.992	*0.994*	0.991	**0.997**	0.990	*0.995*	0.988	0.990	0.992
holder	*0.936*	0.873	*0.913*	*0.941*	*0.930*	**0.965**	*0.949*	0.950	*0.948*	*0.883*	*0.936*
threehumpcamel	**0.988**	*0.981*	*0.978*	*0.970*	0.978	*0.981*	0.949	0.975	0.931	0.971	0.930
rastrigin2	**0.917**	*0.903*	0.882	*0.884*	*0.891*	*0.899*	*0.884*	*0.877*	*0.910*	0.847	0.836
rastrigin4	*0.806*	0.759	0.773	*0.830*	**0.838**	*0.834*	*0.815*	0.769	*0.800*	0.766	0.775
ackley2	0.850	0.772	0.838	0.802	**0.918**	0.832	*0.869*	*0.774*	0.783	0.811	*0.896*
ackley5	0.528	0.557	0.555	0.579	0.562	0.602	0.594	0.604	*0.620*	**0.671**	0.621
levy2	0.925	*0.949*	*0.927*	*0.933*	0.915	0.960	0.961	*0.958*	0.913	**0.963**	0.929
levy3	*0.960*	0.948	*0.962*	*0.954*	*0.962*	*0.951*	0.961	*0.960*	*0.968*	**0.969**	0.951
levy4	*0.968*	*0.959*	*0.970*	*0.970*	*0.974*	*0.962*	*0.950*	*0.976*	**0.976**	*0.970*	*0.972*
griewank2	*0.960*	*0.963*	0.952	*0.958*	**0.966**	*0.954*	0.955	*0.962*	*0.958*	*0.961*	*0.960*
griewank5	0.981	0.984	0.983	*0.985*	*0.984*	*0.985*	0.983	**0.986**	*0.984*	*0.985*	0.983
stybtang2	0.999	0.970	*0.999*	*1.000*	0.999	0.999	0.999	0.999	0.992	**1.000**	0.999
stybtang4	**0.937**	*0.911*	0.897	*0.916*	0.884	*0.915*	0.901	0.900	0.908	0.893	0.883
powell4	*0.976*	*0.965*	*0.973*	*0.975*	*0.972*	*0.977*	0.965	**0.978**	*0.971*	*0.966*	0.957
dixonprice2	*0.988*	*0.985*	**0.990**	*0.989*	*0.963*	*0.967*	*0.953*	*0.959*	*0.945*	*0.982*	0.953
dixonprice4	*0.987*	*0.986*	0.985	*0.958*	0.981	*0.982*	*0.986*	*0.982*	*0.985*	**0.987**	0.971
bukin	0.822	*0.864*	*0.865*	0.844	*0.860*	0.851	*0.861*	0.852	*0.850*	**0.885**	0.826
shekel5	0.273	*0.383*	0.400	*0.414*	*0.413*	*0.402*	*0.405*	*0.425*	0.366	*0.401*	**0.439**
shekel7	0.280	*0.414*	0.330	*0.397*	0.341	*0.380*	0.369	0.378	*0.406*	**0.445**	*0.387*
michal2	0.990	0.999	0.983	*0.977*	*1.000*	*1.000*	*0.982*	*0.967*	*0.984*	**1.000**	*0.961*
Average	0.842	*0.844*	*0.850*	*0.853*	**0.861**	*0.859*	*0.856*	*0.850*	*0.853*	*0.851*	*0.858*

Table 2: Average gap of 100 repeats on all the five “hard” synthetic.

	Rand	EI	2.EI.s	3.EI.s	4.EI.s	10.EI.s	12.EI.s	2.G	3.G	2.R.10	3.R.3
eggholder	0.498	0.613	0.633	0.657	*0.694*	*0.704*	**0.738**	0.583	0.563	0.569	0.518
shubert	0.355	0.408	*0.441*	**0.507**	*0.484*	*0.455*	*0.479*	0.302	0.254	0.271	0.297
bukin	0.600	*0.849*	*0.855*	*0.859*	**0.865**	*0.850*	0.829	0.829	0.811	0.772	0.762
shekel5	0.038	0.286	0.320	0.343	*0.344*	*0.373*	*0.358*	0.265	0.175	**0.378**	*0.350*
shekel7	0.045	0.268	0.313	0.325	0.370	0.358	**0.412**	0.256	0.174	0.376	0.361
Average	0.307	0.485	0.512	0.538	*0.551*	0.548	**0.563**	0.447	0.395	0.473	0.458

Table 3: Average gap of 50 repeats on real functions for all q.EI variants.

	EI	2.EI.b	2.EI.s	3.EI.b	3.EI.s	4.EI.b	4.EI.s	6.EI.b	6.EI.s	8.EI.b	8.EI.s
SVM	0.738	*0.926*	*0.913*	*0.930*	**0.940**	*0.914*	*0.911*	*0.892*	*0.937*	*0.929*	0.834
LDA	0.956	**1.000**	*1.000*	*0.998*	*0.996*	0.996	*0.993*	*0.999*	0.982	*0.995*	*0.995*
LogReg	0.963	*1.000*	*0.998*	*0.999*	*1.000*	*0.999*	*0.999*	**1.000**	*0.999*	*1.000*	*1.000*
NN Boston	*0.470*	*0.491*	*0.467*	*0.490*	*0.478*	*0.495*	*0.460*	*0.460*	**0.502**	*0.455*	*0.467*
NN Cancer	*0.665*	*0.652*	0.627	0.625	*0.654*	*0.640*	*0.686*	0.625	**0.700**	0.609	*0.686*
Robot3d	0.928	*0.959*	*0.960*	*0.944*	*0.962*	*0.956*	0.957	0.960	*0.962*	**0.967**	*0.961*
Robot4d	*0.730*	*0.725*	*0.726*	*0.720*	0.695	**0.764**	0.695	*0.760*	*0.736*	*0.732*	0.697
Average	0.779	*0.821*	*0.813*	*0.815*	*0.818*	*0.823*	*0.815*	*0.813*	**0.831**	*0.812*	*0.806*