Predictive Entropy Search for Multi-objective Bayesian Optimization

Daniel Hernández-Lobato
Universidad Autónoma de Madrid
daniel.hernandez@uam.es

José Miguel Hernández-Lobato
Harvard University
jmhl@seas.harvard.edu

Amar Shah
University of Cambridge
as793@cam.ac.uk

Ryan P. Adams
Harvard University
rpa@seas.harvard.edu

Abstract

We present MPES, a method for multi-objective Bayesian optimization of expensive-to-evaluate black-box functions. At each iteration, MPES chooses an input location to evaluate each objective function on so as to maximally reduce the entropy of the Pareto set of the associated optimization task. The acquisition function employed by MPES is expressed as a sum over the objectives. This enables its use in a *decoupled* scenario, where the different objectives may be evaluated at different input locations in each iteration. Experiments comparing MPES with other related methods from the literature show that it produces significantly better recommendations with a smaller number of evaluations of the objective functions.

1 Introduction

We address the problem of optimizing K real-valued functions $f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})$ over some bounded domain $\mathcal{X} \in \mathbb{R}^d$, where d is the dimensionality of the input space. This is a more general and challenging scenario than the one considered in traditional optimization problems where there is a single-objective function. Specifically, most of the times it is impossible to optimize all the objective functions at the same time since they may be conflicting. An example may be a complex robotic system in which we are interested in minimizing energy consumption and maximizing locomotion speed [1]. Most probably, maximizing locomotion speed will lead to an increase in the energy consumption and vice-versa. In spite of this, it is still possible to find a set of optimal points $\mathcal{X}^\star$ known as the *Pareto set* [2]. Define that $\mathbf{x}$ dominates $\mathbf{x}'$ if $f_k(\mathbf{x}) < f_k(\mathbf{x}')\ \forall k$. Then, the Pareto set is the subset of non-dominated points in $\mathcal{X}$. Namely, $\forall \mathbf{x}^\star \in \mathcal{X}^\star, \forall \mathbf{x} \in \mathcal{X}, \exists\, k \in \{1, \ldots, K\}$ such that $f_k(\mathbf{x}^\star) \leq f_k(\mathbf{x})$, assuming minimization. Given $\mathcal{X}^\star$, the final user may choose a point from this set according to their preferences (*e.g.*, locomotion speed vs. energy consumption). The Pareto set is often not finite, and most strategies aim at finding a finite set to approximate $\mathcal{X}^\star$ well.

Frequently, the cost of evaluating each function $f_k(\cdot)$ is notoriously very high. In this case, one attempts to minimize the number of evaluations required to obtain the final approximation to the Pareto set $\mathcal{X}^\star$. An approach that has shown promising results in such a setting consists in using probabilistic models (typically a Gaussian process) to approximate the output of each function [3, 4, 5, 6]. At each iteration, these strategies use the uncertainty in the probabilistic models to generate an acquisition function whose maximum indicates the most promising location on which to evaluate the objectives. This contrasts with model-free methods based on genetic algorithms or evolutionary strategies that are very effective for approximating the Pareto set, but demand a large number of function evaluations [7, 8, 9]. A limitation of current model-based approaches is, however, that (i) they often build the acquisition function by transforming the multi-objective problem into a single-objective problem using scalarization techniques (an approach that is expected to be suboptimal), and (ii) the acquisition function cannot be expressed as a sum over the different objectives, which enforces the evaluation of all objective functions at the exact same input location in each iteration.

We describe here a strategy for multi-objective optimization that is suited to the scenario described. For this, we extend previous single-objective strategies based on stepwise uncertainty reduction to the multi-objective case [10, 11, 12]. In the single-objective case, these strategies choose the next evaluation location based on the reduction of the Shannon entropy of the minimizer $\mathbf{x}^\star$. The idea is that a smaller entropy implies that the minimizer $\mathbf{x}^\star$ is better-identified. Furthermore, such a criterion has been shown to provide better results than other alternatives based, *e.g.*, on the *expected improvement* [11, 12]. The extension to the multi-objective case is obtained by considering the entropy of the Pareto set $\mathcal{X}^\star$. More precisely, we choose the next evaluation as the one that is expected to reduce the most the entropy of $\mathcal{X}^\star$. The proposed approach is called *Multi-objective Predictive Entropy Search* (MPES). Several experiments show that MPES leads to better performance than related methods from the literature, especially with noisy evaluations. Furthermore, in MPES the acquisition function is expressed as a sum across the different objectives. This allows for a *decoupled* scenario in which the objectives may be evaluated at different input locations in each iteration.

2 Multi-objective Bayesian optimization via predictive entropy search

Given some evaluations of each objective function $f_k(\cdot)$ we seek to maximize the information about the Pareto set $\mathcal{X}^\star$. For this, we assume that each $f_k(\cdot)$ follows a Gaussian process (GP) prior [13], with potential observation noise that is i.i.d Gaussian with zero mean. GPs are often used in model-based approaches for multi-objective optimization because of their flexibility and intrinsic ability to model uncertainty [3, 4, 5, 6]. For simplicity, we consider a coupled setting in which we evaluate each objective at the same location. Nevertheless, the approach described can be easily extended to the *decoupled* scenario. Let $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ be the data collected up to step N, where $\mathbf{y}_n$ is a K-dimensional vector with the values resulting from the evaluation of all the objectives at step n, and $\mathbf{x}_n$ is a vector in input space denoting the evaluation location. The next query $\mathbf{x}_{N+1}$ is defined as the one that maximizes the expected reduction in the differential entropy $H(\cdot)$ of the posterior distribution of the Pareto set $\mathcal{X}^\star$, $p(\mathcal{X}^\star|\mathcal{D})$. More precisely, the acquisition function of MPES is:

$$\alpha(\mathbf{x}) = H(\mathcal{X}^\star|\mathcal{D}) - \mathbb{E}_{\mathbf{y}}\left[H(\mathcal{X}^\star|\mathcal{D} \cup \{(\mathbf{x}, \mathbf{y})\})\right] , \tag{1}$$

where $\mathbf{y}$ is the output of all the GP models at $\mathbf{x}$ and the expectation is taken with respect to the posterior distribution for $\mathbf{y}$ given by these models, $p(\mathbf{y}|\mathcal{D}, \mathbf{x}) = \prod_{k=1}^K p(y_k|\mathcal{D}, \mathbf{x})$, under the assumption of independence among the different GPs. Thus, at each iteration we set $\mathbf{x}_{N+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x})$.

A practical difficulty is, however, that the exact evaluation of (1) is infeasible. This function has to be approximated. For this, we follow the approach described in [11, 14]. In particular, (1) is the mutual information between $\mathcal{X}^\star$ and $\mathbf{y}$ given $\mathcal{D}$. The mutual information is symmetric and hence we can exchange the roles of the variables $\mathcal{X}^\star$ and $\mathbf{y}$. This leads to the equivalent expression to (1):

$$\alpha(\mathbf{x}) = H(\mathbf{y}|\mathcal{D}, \mathbf{x}) - \mathbb{E}_{\mathcal{X}^\star}\left[H(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)\right] , \tag{2}$$

where the expectation is now with respect to the posterior distribution for the Pareto set $\mathcal{X}^\star$ given the observed data, and $H(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)$ measures the entropy of $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)$, *i.e.*, the predictive distribution for the objectives at $\mathbf{x}$ given $\mathcal{D}$ and conditioned to $\mathcal{X}^\star$ being the Pareto set of the objective functions. This alternative formulation significantly simplifies the evaluation of the acquisition function $\alpha(\cdot)$. In particular, we no longer have to evaluate the entropy of $\mathcal{X}^\star$, which may be difficult. The new acquisition function obtained in (2) favors the regions of the input space in which $\mathcal{X}^\star$ is more informative about $\mathbf{y}$, and these are also the regions in which $\mathbf{y}$ is more informative about $\mathcal{X}^\star$.

The first term in the r.h.s. of (2) is straight-forward to evaluate. It is simply the entropy of the predictive distribution $p(\mathbf{y}|\mathcal{D}, \mathbf{x})$, which is a factorizing K-dimensional Gaussian distribution. Thus, $H(\mathbf{y}|\mathcal{D}, \mathbf{x}) = \frac{K}{2}\log(2\pi e) + \sum_{i=1}^K 0.5\log(v_k^{\text{PD}})$, where v_k^{PD} is the predictive variance of $f_k(\cdot)$ at $\mathbf{x}$. The difficulty comes from the evaluation of the second term in the r.h.s. of (2), which is intractable and has to be approximated. For this, we follow [11] and approximate the expectation using a Monte Carlo estimate obtained by drawing samples of $\mathcal{X}^\star$ given $\mathcal{D}$. This involves sampling several times the objective functions[1] from the posterior $p(f_1, \ldots, f_K|\mathcal{D})$. This is done as in [11]. Unlike the true objectives, the sampled functions can be evaluated without significant cost. Thus, given these functions, we use a grid search with $d \times 1,000$ points to find $\mathcal{X}^\star$ (in higher dimensions the NSGA-II evolutionary algorithm may be preferred [7]), which is then approximated in our case using a representative sub-set of 50 points. Given a sample of $\mathcal{X}^\star$, the differential entropy of $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)$ is estimated using expectation propagation [15], as described in the next section.

[1] We generate 10 samples from the posterior distribution of each objective function.

2.1 Approximating the conditional predictive distribution using expectation propagation

To approximate the entropy of the conditional predictive distribution $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)$ we consider the distribution $p(\mathcal{X}^\star|f_1, \ldots, f_K)$. In particular, $\mathcal{X}^\star$ is the Pareto set of $f_1, \ldots, f_K$ i.f.f. $\forall \mathbf{x}^\star \in \mathcal{X}^\star, \forall \mathbf{x}' \in \mathcal{X}, \exists\, k \in \{1, \ldots, K\}$ such that $f_k(\mathbf{x}^\star) \leq f_k(\mathbf{x}')$, assuming minimization. Informally, the conditions just described can be translated into the following un-normalized distribution:

$$p(\mathcal{X}^\star|f_1, \ldots, f_K) \propto \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \prod_{\mathbf{x}' \in \mathcal{X}} \left[1 - \prod_{k=1}^{K} \Theta\left(f_k(\mathbf{x}') - f_k(\mathbf{x}^\star)\right)\right] = \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \prod_{\mathbf{x}' \in \mathcal{X}} \psi(\mathbf{x}', \mathbf{x}^\star)\,, \quad (3)$$

where $\psi(\mathbf{x}', \mathbf{x}^\star) = 1 - \prod_{k=1}^{K} \Theta\left(f_k(\mathbf{x}') - f_k(\mathbf{x}^\star)\right)$, $\Theta(\cdot)$ is the Heaviside step function, and we have used the convention that $\Theta(0) = 1$. Thus, the r.h.s. of (3) is different from zero only for a valid Pareto set. Next, we note that $p(\mathbf{y}|\mathbf{x}, f_1, \ldots, f_K) = \prod_{i=1}^{K} \delta(y_k - f_k(\mathbf{x}))$, with $\delta(\cdot)$ the Dirac's delta function. This means that we can write the un-normalized version of $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)$ as follows:

$$\begin{aligned} p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) &\propto \int p(\mathbf{y}|\mathbf{x}, f_1, \ldots, f_K) p(\mathcal{X}^\star|f_1, \ldots, f_K) p(f_1, \ldots, f_K|\mathcal{D}) df_1 \cdots df_K \\ &\propto \int \prod_{i=1}^{K} \delta(y_k - f_k(\mathbf{x})) \prod_{\mathbf{x}^\star \in \mathcal{X}^\star} \psi(\mathbf{x}, \mathbf{x}^\star) \left[\prod_{\mathbf{x}' \in \mathcal{X} \setminus \{\mathbf{x}\}} \psi(\mathbf{x}', \mathbf{x}^\star)\right] \\ &\quad \times p(f_1, \ldots, f_K|\mathcal{D}) df_1 \cdots df_K\,, \end{aligned} \quad (4)$$

where we have separated out the factors ψ that do not depend on $\mathbf{x}$. The approximation to the r.h.s. of (4) is obtained in two stages. First, we approximate $\mathcal{X}$ with the set $\tilde{\mathcal{X}} = \{\mathbf{x}_n\}_{n=1}^{N} \cup \mathcal{X}^\star$, *i.e.*, the union of the input locations where the objective functions have been already evaluated and the current Pareto set. Then, we replace each non-Gaussian factor ψ with a corresponding approximate Gaussian factor $\tilde{\psi}$ whose parameters are adjusted using expectation propagation (EP) [15]. That is,

$$\psi(\mathbf{x}', \mathbf{x}^\star) = 1 - \prod_{k=1}^{K} \Theta\left(f_k(\mathbf{x}') - f_k(\mathbf{x}^\star)\right) \approx \tilde{\psi}(\mathbf{x}', \mathbf{x}^\star) = \prod_{k=1}^{K} \tilde{\phi}_k(f_k(\mathbf{x}'), f_k(\mathbf{x}^\star))\,, \quad (5)$$

where each factor $\tilde{\phi}_k$ is an un-normalized two-dimensional Gaussian distribution. In particular, $\tilde{\phi}_k(f_k(\mathbf{x}'), f_k(\mathbf{x}^\star)) = \exp\{-0.5\boldsymbol{\upsilon}_k^{\mathrm{T}} \tilde{\mathbf{V}}_k \boldsymbol{\upsilon}_k + \tilde{\mathbf{m}}_k^{\mathrm{T}} \boldsymbol{\upsilon}_k\}$, where we have defined $\boldsymbol{\upsilon}_k = (f_k(\mathbf{x}'), f_k(\mathbf{x}^\star))^{\mathrm{T}}$ and $\tilde{\mathbf{V}}_k$, and $\tilde{\mathbf{m}}_k$ are parameters to be adjusted by EP, which refines each $\tilde{\psi}$ until convergence to enforce that it looks similar to the corresponding exact factor ψ [16]. The approximate factors $\tilde{\psi}$ that do not depend on the candidate input $\mathbf{x}$ are reused multiple times to evaluate the acquisition function $\alpha(\mathbf{x})$, and they only have to be computed once. The $|\mathcal{X}^\star|$ factors that depend on $\mathbf{x}$ must be obtained relatively fast. For this reason, we only update these factors once using EP.

Once EP has been run, we approximate $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)$ by the normalized Gaussian that results from replacing each exact factor ψ by the corresponding approximate one $\tilde{\psi}$. That is, $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star) \approx \prod_{i=1}^{K} \mathcal{N}(f_k(\mathbf{x})|m_k^{\mathrm{CPD}}, v_k^{\mathrm{CPD}})$, where the parameters m_k^{CPD} and v_k^{CPD} can be obtained from each $\tilde{\psi}$ and $p(f_1, \ldots, f_K|\mathcal{D})$. This leads to the following approximation of the acquisition function in (2):

$$\alpha(\mathbf{x}) \approx \sum_{k=1}^{K} 0.5 \log v_k^{\mathrm{PD}}(\mathbf{x}) - \frac{1}{S} \sum_{s=1}^{S} \sum_{k=1}^{K} 0.5 \log v_k^{\mathrm{CPD}}(\mathbf{x}|\mathcal{X}_{(s)}^\star)\,, \quad (6)$$

where $\{\mathcal{X}_{(s)}^\star\}_{s=1}^{S}$ are the Pareto sets sampled to approximate the expectation in (2), and $v_k^{\mathrm{PD}}(\mathbf{x})$ and $v_k^{\mathrm{CPD}}(\mathbf{x}|\mathcal{X}_{(s)}^\star)$ are respectively the variances of the predictive distribution for $f_1, \ldots, f_K$ at $\mathbf{x}$, before and after conditioning to $\mathcal{X}_{(s)}^\star$. Furthermore, if additive Gaussian noise is considered around the objective functions, we just add the variance of the noise to the corresponding variances. Finally, the total computational cost of evaluating the acquisition function $\alpha(\mathbf{x})$ is $\mathcal{O}(m^3)$, where $m = N + |\mathcal{X}^\star|$.

3 Experiments

We compare MPES with other strategies from the literature. Namely, ParEGO [3], SMSego [5], EHI [4] and SUR [6]. ParEGO transforms the multi-objective problem into a single-objective problem using a scalarization technique. Given the scalarization, the acquisition function uses the expected improvement [17, 18]. SMSego computes the gain in the hyper-volume indicator (which is used to measure performance) obtained from a lower-confidence bound on the values of the objectives.

EHI uses the expected improvement of the hyper-volume at each location to choose the next point. Finally, SUR computes the reduction of the volume of excursion behind the Pareto front. We coded all methods in Spearmint[2]. In all GP models we used a Matérn covariance function. All hyper-parameters are approximately sampled from their posterior and the acquisition function of each method is averaged over these samples. At iteration n, each method gives a recommendation in the form of a Pareto set obtained by optimizing the posterior means.

Accuracy of the approximation of the acquisition function: We compare in a one-dimensional problem with two objectives the acquisition function computed by MPES with a more accurate estimate obtained via expensive Monte Carlo sampling and a non-parametric estimator of the entropy [19]. Figure 1 (left) shows at a given step the observed data and the posterior mean and the standard deviation of each objective. The right figure shows the corresponding acquisition function computed by MPES and by the Monte Carlo method (Exact). We observe that both functions look very similar, including the location of the global maximizer. This shows that (6) is a good approximation of (2).

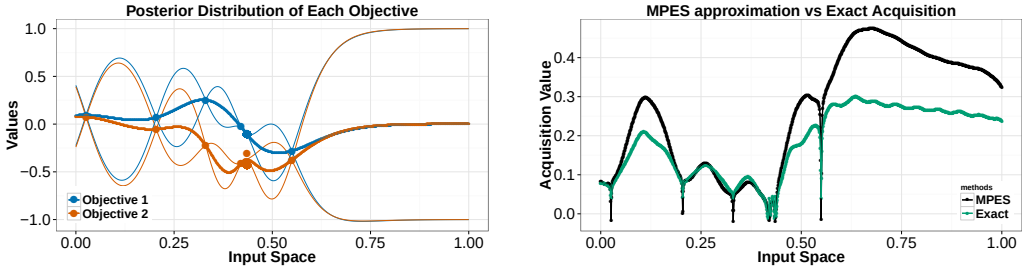

Figure 1: (left) Observations of each objective and posterior mean and standard deviations of each GP model. (right) Estimates of the acquisition function (2) by MPES, and by a Monte Carlo method combined with a non-parametric estimator of the entropy (Exact), which is expected to be more accurate. Best seen in color.

Experiments with synthetic data: Consider a three-dimensional problem with two objectives obtained by sampling the functions from the corresponding GP prior. We generate 100 of these problems and report the average performance of each method on them, when considering no-noise around the observations and when the observations are contaminated with additive Gaussian noise with standard deviation equal to 0.1. The performance metric employed is the hyper-volume indicator, which is maximized by the actual Pareto set [20]. Figure 2 shows, at each iteration, the performance of the recommendations made, computed in terms of the log relative difference between the hyper-volume of the recommendation and the highest possible hyper-volume. MPES obtains the best results. When executed in a decoupled scenario, it gives slightly improvements in the case of noisy observations.

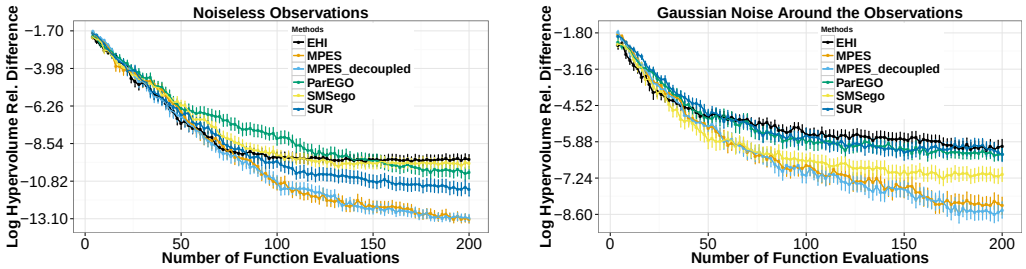

Figure 2: (left) Average log relative difference between the hyper-volume of the recommendation and the maximum hyper-volume for each number of evaluations made. (right) Similar results are obtained when the evaluations of the objective functions are contaminated with additive Gaussian noise. Best seen in color.

4 Conclusions and Future Work

We have described MPES, a method for multi-objective Bayesian optimization. At each iteration, MPES evaluates the objective functions at the input location that is expected to reduce the entropy of the Pareto set the most. Synthetic experiments show that MPES has better performance than other methods. That is, MPES obtains better recommendations with a smaller number of evaluations, both in the case of noiseless and noisy observations. Finally, MPES's acquisition function involves a sum over the objectives. This allows for a *decoupled* evaluation scenario. Future work includes considering real-world multi-objective problems and problems with more than two objectives.

[2] `https://github.com/JasperSnoek/spearmint`

Acknowledgments: JMHL acknowledges support from the Rafael del Pino Foundation. DHL and JMHL acknowledge support from Plan Nacional I+D+i, Grant TIN2013-42351-P, and from Comunidad Autónoma de Madrid, Grant S2013/ICE-2845 CASI-CAM-CM. DHL is grateful for using the computational resources of *Centro de Computación Científica* at Universidad Autónoma de Madrid. AS acknowledges useful discussions with Zoubin Ghahramani.

References

[1] R. Ariizumi, M. Tesch, H. Choset, and F. Matsuno. Expensive multiobjective optimization for robotics with consideration of heteroscedastic noise. In *2014 IEEE International Conference on Intelligent Robots and Systems*, pages 2230–2235, 2014.

[2] Y. Collette and P. Siarry. *Multiobjective Optimization: Principles and Case Studies*. Springer, 2003.

[3] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10:50–66, 2006.

[4] A Emmerich. The computation of the expected improvement in dominated hypervolume of Pareto front approximations. Technical Report LIACS TR-4-2008, Leiden University, The Netherlands, 2008.

[5] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection. In *Parallel Problem Solving from Nature PPSN X*, pages 784–794. 2008.

[6] V. Picheny. Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25:1265–1280, 2015.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[8] X. Li. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Genetic and Evolutionary Computation GECCO 2003*, pages 37–48. 2003.

[9] M. Laumanns E. Zitzler and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, 2001.

[10] J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44:509–534, 2009.

[11] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems 27*, pages 918–926. 2014.

[12] Philipp H. and Christian J. S. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13:1809–1837, 2012.

[13] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

[14] N. Houlsby, J. M. Hernández-lobato, F. H., and Z. Ghahramani. Collaborative gaussian processes for preference learning. In *Advances in Neural Information Processing Systems 25*, pages 2096–2104. 2012.

[15] T. Minka. Expectation propagation for approximate Bayesian inference. In *Annual Conference on Uncertainty in Artificial Intelligence*, pages 362–36, 2001.

[16] T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT, 2001.

[17] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[18] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[19] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23:301–321, 2003.

[20] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3:257–271, 1999.