Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints

Eduardo C. Garrido-Merchán
Universidad Autónoma de Madrid
eduardo.garrido@uam.es

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

Abstract

We present PESMOC, Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints, an information-based strategy for the simultaneous optimization of multiple expensive-to-evaluate black-box objectives under the presence of constraints. Iteratively, PESMOC chooses an input location on which to evaluate the objectives and the constraints so as to maximally reduce the entropy of the Pareto set of the optimization problem. The constraints are assumed to have similar properties to the objectives in typical Bayesian optimization problems. They do not have a known expression, their evaluation is very expensive, and the observations may be noisy. Several synthetic experiments illustrate the effectiveness of PESMOC and compare its performance with state-of-the-art methods.

1 Introduction

We consider the problem of simultaneously minimizing K functions $f_1(\mathbf{x}), ..., f_K(\mathbf{x})$ subject to the non-negativity of C constraints $c_1(\mathbf{x}), ..., c_C(\mathbf{x})$, over some bounded domain $\mathcal{X} \in \mathbb{R}^d$, where d is the dimensionality of the input space. More precisely, the problem considered is:

$$\min_{\mathbf{x}\in\mathcal{X}} \quad f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}) \qquad\qquad \text{s.t.} \qquad c_1(\mathbf{x}) \geq 0, \ldots, c_C(\mathbf{x}) \geq 0\,.$$

We say that a point $\mathbf{x} \in \mathcal{X}$ is feasible if $c_j(\mathbf{x}) \geq 0, \forall j$. The feasible space $\mathcal{F} \in \mathcal{X}$ is hence the set of points that are feasible. Only the solutions contained in $\mathcal{F}$ are considered valid. In practice, however, most of the times it is impossible to optimize all the objective functions at the same time, as they may be conflicting. An example is the control system of a four-legged robot in which we need to minimize energy consumption and maximize locomotion speed [1]. Most probably, maximizing locomotion speed will lead to an increase in the energy consumption and minimizing the energy consumption will decrease locomotion speed. In spite of this, it is still possible to find a set of optimal points $\mathcal{X}^\star$ known as the *Pareto set* [2]. Let's define that the point $\mathbf{x}$ dominates the point $\mathbf{x}'$ if $f_k(\mathbf{x}) \leq f_k(\mathbf{x}')$ $\forall k$, with at least one inequality being strict. Then, the Pareto set is the subset of non-dominated points in $\mathcal{F}$. Namely, $\forall \mathbf{x}^\star \in \mathcal{X}^\star \subset \mathcal{F}, \forall \mathbf{x} \in \mathcal{F}\, \exists\, k \in 1, ..., K$ such that $f_k(\mathbf{x}^\star) < f_k(\mathbf{x})$. Typically, given $\mathcal{X}^\star$ the final user may choose a point from this set according to their needs (locomotion speed vs. energy consumption). Note that several constraints may also appear in such an optimization problem. For example, the amount of weight placed on a leg of the robot should not exceed a particular value. Another illustrative example includes the optimization of a system for object recognition. We may have a deep neural network and we would like to maximize prediction accuracy and minimize prediction time under the constraint that when codifying such network into a chip the energy consumption is below a particular level, so that the chip can be included in a mobile device.

In the problems described, the cost of evaluating the objectives and the constraints may be very high, the evaluations can be noisy and is unlikely to find closed form expressions for these functions, which make difficult any gradient computation. An approach that has shown promising results in such an optimization setting is Bayesian optimization (BO) [3]. In BO a probabilistic model (typically a

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Gaussian process, GP, [4]) is used to describe the output of each function. At each iteration, these techniques use the models to generate an acquisition function whose maximum indicates the most promising location on which to evaluate the functions. After enough observations have been collected, the models can be optimized to provide an estimate of the Pareto set. Most times, BO methods find a good estimate of the solution of the problem with a small number of evaluations [5, 6].

In this work we describe a strategy for constrained multi-objective optimization. We extend previous work that uses information theory to optimize several objectives [7] or a single objective with several constraints [8]. The result is a strategy that can handle several objectives and several constraints at the same time. This strategy chooses the next point on which to evaluate the objectives and the constraints as the one that is expected to reduce the most the uncertainty about the Pareto set in the feasible space, measured in terms of *Shannon's differential entropy*. A smaller entropy implies that the Pareto set is better-identified [9, 10, 11]. The proposed approach is called Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints (PESMOC). We show, using experiments where both the objectives and the constraints are sampled from a GP prior, that PESMOC has practical advantages over a random search strategy and that provides better Pareto set recommendations than a state-of-the-art approach based on the expected hypervolume improvement [12].

2 Predictive Entropy Search for Multi-objective Optimization with Constraints

We maximize the information gain about the Pareto set $\mathcal{X}^\star$ over the feasible set $\mathcal{F}$. Let the objectives $\{f_1, \ldots, f_K\}$ be denoted with $\mathbf{f}$ and the constraints $\{c_1, \ldots, c_C\}$ be denoted with $\mathbf{c}$. We assume that they have been generated from independent GP priors [4]. A coupled setting in which all functions are evaluated at the same location at each iteration is considered. Let $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ be all the observations collected up to step N, where $\mathbf{y}_n$ is a $K + C$-dimensional vector with the evaluations of objectives and the constraints at step n, and $\mathbf{x}_n$ is the input location. The next evaluation point $\mathbf{x}_{N+1}$ is the one that maximizes the expected reduction in the differential entropy $H(\cdot)$ of the posterior distribution over the Pareto set, $p(\mathcal{X}^\star|\mathcal{D})$. More precisely, the PESMOC acquisition function is:

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

where the expectation is taken with respect to the posterior distribution of the potentially noisy evaluations of the objectives $\mathbf{f}$ and the constraints $\mathbf{c}$, at $\mathbf{x}$. That is, $p(\mathbf{y}|\mathcal{D}, \mathbf{x}) = \prod_{k=1}^K p(y_k|\mathcal{D}, \mathbf{x}) \prod_{j=1}^C p(y_{K+j}|\mathcal{D}, \mathbf{x})$. Note that the computation of (1), known as *Entropy Search* [10], is very difficult since it involves the entropy of a set of points of potentially infinite size. Following [13, 11] the computation can be made easier by noting that (1) is the mutual information between $\mathcal{X}^\star$ and $\mathbf{y}$. The mutual information is symmetric and hence the roles of $\mathcal{X}^\star$ and $\mathbf{y}$ can be swapped:

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

where the expectation is now with respect to the posterior distribution for the Pareto set, $\mathcal{X}^\star$, given the observed data, $\mathcal{D}$, 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 and the constraints at $\mathbf{x}$ given $\mathcal{D}$ and conditioned to $\mathcal{X}^\star$ being the Pareto set. Note that the first term in the r.h.s. of (2) is easy to evaluate. It is simply the entropy of the predictive distribution $p(\mathbf{y}|\mathcal{D}, \mathbf{x})$, a factorizing $K + C$-dimensional Gaussian distribution: $H(\mathbf{y}|\mathcal{D}, \mathbf{x}) = \frac{K+C}{2} \log(2\pi e) + \sum_{i=1}^K 0.5 \log(v_k^{\text{PD}}) + \sum_{i=1}^C 0.5 \log(s_c^{\text{PD}})$ where v_k^{PD} and s_c^{PD} are the predictive variances of the objectives and the constraints, respectively. However, the second term in (2) has to be approximated. We use a Monte Carlo estimate obtained by drawing samples of $\mathcal{X}^\star$ given $\mathcal{D}$. For this, we sample the objectives and the constraints from their posterior $p(\mathbf{f}, \mathbf{c}|\mathcal{D})$ following [11, 7]. Given these samples, we solve the corresponding optimization problem to get a sample of $\mathcal{X}^\star$. Because the samples of $\mathbf{f}$ and $\mathbf{c}$ can be evaluated very quickly, this step has little cost. Given a sample of $\mathcal{X}^\star$, the entropy of $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)$ is estimated using expectation propagation [14].

2.1 Using Expectation Propagation to Approximate the Conditional Predictive Distribution

We use expectation propagation (EP) to approximate the entropy of the conditional predictive distribution (CPD) $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathcal{X}^\star)$ [14]. For this, the distribution $p(\mathcal{X}^\star|\mathbf{f}, \mathbf{c})$ is considered first. Note that $\mathcal{X}^\star$ is the Pareto set in $\mathcal{F}$ if and only if $\forall \mathbf{x}^\star \in \mathcal{X}^\star$, $\forall \mathbf{x}' \in \mathcal{X}$, $c_j(\mathbf{x}^\star) \geq 0 \; \forall j$, and if $c_j(\mathbf{x}') \geq 0$,

$\forall j$, then $\exists k$ s.t. $f_k(\mathbf{x}^\star) < f_k(\mathbf{x}')$. These conditions can be informally summarized as:

$$p(\mathcal{X}^\star|\mathbf{f},\mathbf{c}) \propto \prod_{\mathbf{x}^\star\in\mathcal{X}^\star}\left(\left[\prod_{j=1}^{C}\Phi_j(\mathbf{x}^\star)\right]\left[\prod_{\mathbf{x}'\in\mathcal{X}}\Omega(\mathbf{x}',\mathbf{x}^\star)\right]\right)\,, \tag{3}$$

where we have defined $\Omega(\mathbf{x}',\mathbf{x}^\star) = \left[\prod_{j=1}^{C}\Theta(c_j(\mathbf{x}'))\right]\psi(\mathbf{x}',\mathbf{x}^\star) + \left[1 - \prod_{j=1}^{C}\Theta(c_j(\mathbf{x}'))\right]\cdot 1$, $\psi(\mathbf{x}',\mathbf{x}^\star) = 1 - \prod_{k=1}^{K}\Theta(f_k(\mathbf{x}^\star) - f_k(\mathbf{x}'))$ and $\Phi_j(\mathbf{x}^\star) = \Theta(c_j(\mathbf{x}^\star))$, with $\Theta(\cdot)$ the Heaviside step function. The factor $\prod_{j=1}^{C}\Phi_j(\mathbf{x}^\star)$ in (3) guarantees that every point in $\mathcal{X}^\star$ belongs to $\mathcal{F}$. Similarly, the product $\prod_{j=1}^{C}\Theta(c_j(\mathbf{x}'))$ checks that $\mathbf{x}'$ belongs to $\mathcal{F}$. Note that if $\mathbf{x}'$ is not feasible, we do nothing, *i.e.*, we multiply by one. Otherwise, $\mathbf{x}'$ has to be dominated by $\mathbf{x}^\star$. The factor $\psi(\mathbf{x}',\mathbf{x}^\star)$ checks that. In summary, the r.h.s. of (3) takes value one if $\mathcal{X}^\star$ is a valid Pareto set and zero otherwise. $\mathcal{X}^\star$ is a random variable as we have a probability distribution over the objectives and constraints.

Now we show how to approximate $p(\mathbf{y}|\mathcal{D},\mathbf{x},\mathcal{X}^\star)$. For simplicity, we consider a noiseless case in which we observe the actual objectives and constraints: $p(\mathbf{y}|\mathbf{x},\mathbf{f},\mathbf{c}) = \prod_{k=1}^{K}\delta(y_k - f_k(\mathbf{x}))\prod_{j=1}^{C}\delta(y_{K+j} - c_j(\mathbf{x}))$, where $\delta(\cdot)$ is a Dirac's delta function (in the noisy case we simply use Gaussians). The unnormalized version of $p(\mathbf{y}|\mathcal{D},\mathbf{x},\mathcal{X}^\star)$ is:

$$p(\mathbf{y}|\mathcal{D},\mathbf{x},\mathcal{X}^\star) \propto \int p(\mathbf{y}|\mathbf{x},\mathbf{f},\mathbf{c})p(\mathcal{X}^\star|\mathbf{f},\mathbf{c})p(\mathbf{f}|\mathcal{D})p(\mathbf{c}|\mathcal{D})d\mathbf{f}d\mathbf{c} \propto \int \prod_{k=1}^{K}\delta(y_k - f_k(\mathbf{x}))\prod_{j=1}^{C}\delta(y_{K+j} - c_j(\mathbf{x}))\times$$

$$\prod_{\mathbf{x}^\star\in\mathcal{X}^\star}\prod_{j=1}^{C}\Phi_j(\mathbf{x}^\star)\times\prod_{\mathbf{x}^\star\in\mathcal{X}^\star}\left(\Omega(\mathbf{x},\mathbf{x}^\star)\prod_{\mathbf{x}'\in\mathcal{X}\setminus\{\mathbf{x}\}}\Omega(\mathbf{x}',\mathbf{x}^\star)\right)\times p(\mathbf{f}|\mathcal{D})p(\mathbf{c}|\mathcal{D})d\mathbf{f}d\mathbf{c}\,, \tag{4}$$

To approximate (4) $\mathcal{X}$ is replaced by $\hat{\mathcal{X}} = \{\mathbf{x}_n\}_{n=1}^{N}\cup\mathcal{X}^\star\cup\{\mathbf{x}\}$, where $\{\mathbf{x}_n\}_{n=1}^{N}$ is the union of the input locations where the functions have been evaluated. All non-Gaussian factors in (4) are then replaced by Gaussian approximate factors using EP. Each $\Phi_j(\cdot)$ factor is replaced by an un-normalized Gaussian distribution over $c_j(\mathbf{x}^\star)$: $\Phi_j(\mathbf{x}^\star) \approx \tilde{\Phi}_j(\mathbf{x}^\star) \propto \exp\{-0.5\cdot c_j(\mathbf{x}^\star)^2\tilde{v}_j^{\mathbf{x}^\star} + c_j(\mathbf{x}^\star)\tilde{m}_j^{\mathbf{x}^\star}\}$, where $\tilde{v}_j^{\mathbf{x}^\star}$ and $\tilde{m}_j^{\mathbf{x}^\star}$ are natural parameters. Similarly, each $\Omega(\mathbf{x}',\mathbf{x}^\star)$ factor is replaced by a product of C one-dimensional and K two-dimensional un-normalized Gaussians: $\Omega(\mathbf{x}',\mathbf{x}^\star) \approx \tilde{\Omega}(\mathbf{x}',\mathbf{x}^\star) \propto \prod_{k=1}^{K}\exp\{-0.5\cdot\boldsymbol{\upsilon}_k^{\mathrm{T}}\tilde{\mathbf{V}}_k^{\Omega}\boldsymbol{\upsilon}_k + (\tilde{\mathbf{m}}_k^{\Omega})^{\mathrm{T}}\boldsymbol{\upsilon}_k\}\times\prod_{j=1}^{C}\exp\{-0.5\cdot c_j(\mathbf{x}^\star)^2\tilde{v}_j^{\Omega} + c_j(\mathbf{x}^\star)\tilde{m}_j^{\Omega}\}$, where $\boldsymbol{\upsilon}_k = (f_k(\mathbf{x}'), f_k(\mathbf{x}^\star))^{\mathrm{T}}$ and $\tilde{\mathbf{V}}_k^{\Omega}$, $\tilde{\mathbf{m}}_k^{\Omega}$, $\tilde{v}_j^{\Omega}$ and $\tilde{m}_j^{\Omega}$ are natural parameters. EP adjust all them. The factors, that do not depend on $\mathbf{x}$ are refined iteratively by EP until they do not change, and are reused each time that the acquisition function has to be computed at a new point $\mathbf{x}$. When EP finishes, $p(\mathbf{y}|\mathcal{D},\mathbf{x},\mathcal{X}^\star)$ is approximated by the Gaussian distribution that results by replacing in (4) each non-Gaussian factor by the corresponding EP Gaussian approximation. The result is an un-normalized Gaussian distribution. The PESMOC acquisition function is hence:

$$\alpha(\mathbf{x}) \approx \sum_{j=1}^{C}\log s_j^{\mathrm{PD}}(\mathbf{x}) + \sum_{k=1}^{K}\log v_k^{\mathrm{PD}}(\mathbf{x}) - \frac{1}{M}\sum_{m=1}^{M}\left[\sum_{j=1}^{C}\log s_j^{\mathrm{CPD}}(\mathbf{x}|\mathcal{X}^\star_{(m)}) + \sum_{k=1}^{K}\log v_k^{\mathrm{CPD}}(\mathbf{x}|\mathcal{X}^\star_{(m)})\right]\,, \tag{5}$$

where M is the number of samples used to approximate the expectation in the r.h.s. of (2), and $v_k^{\mathrm{PD}}(\mathbf{x})$, $v_c^{PD}(\mathbf{x})$, $v_k^{\mathrm{CPD}}(\mathbf{x}|\mathcal{X}^\star_{(m)})$ and $v_c^{\mathrm{CPD}}(\mathbf{x}|\mathcal{X}^\star_{(m)})$ are the variances of the predictive distribution before and after conditioning on the Pareto set. The cost of evaluating the acquisition function is $\mathcal{O}((K+C)q^3)$, with $q = N + |\mathcal{X}^\star_{(m)}|$, and N the number of observations. In practice EP is run only once per sample of the Pareto set $\mathcal{X}^\star_{(m)}$ because it is possible to re-use the factors that are independent of the candidate location $\mathbf{x}$. Thus, the complexity of computing the predictive variance is $\mathcal{O}((K+C)|\mathcal{X}^\star_{(s)}|^3)$. PESMOC scales linearly with respect to the number of black-boxes. It can hence address complicated problems with a large number of objectives and constraints.

3 Experiments

We carry out experiments to compare the performance of PESMOC with a random search (RS) strategy and a method based on the expected hypervolume improvement (BMOO) [12]. All the strategies have been implemented in the software for Bayesian optimization Spearmint (`https://github.com/HIPS/Spearmint`). A Matérn covariance function is used in the GPs that model the objectives and the constraints. The hyper-parameters of each GPs are approximately sampled from their posterior distribution using slice sampling [15]. We generate 10 samples for each hyper-parameter, and the acquisition function of PESMOC and BMOO is averaged over these samples.

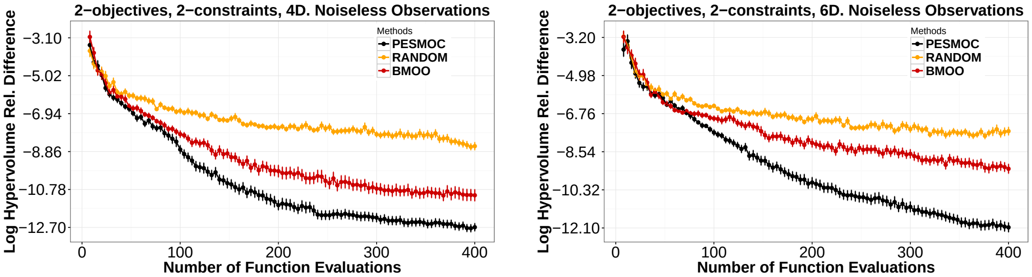

Figure 1: Logarithm of the relative difference between the hyper-volume of the recommendation obtained by the three methods. We report results after each evaluation of the functions. (left) Input space of four dimensions ($d = 4$). (right) Input space of six dimensions ($d = 6$). Best seen in color.

We generate 100 optimization problems by sampling 2 objectives and 2 constraints from a GP prior. We consider two values for the input dimension: $d = 4$ and $d = 6$. Each strategy (PESMOC, BMOO and RS) is then run on each problem until 100 evaluations of each function are made. For the sake of simplicity we consider a noiseless scenario. After each iteration, each strategy outputs a recommendation in the form of a Pareto set obtained by optimizing the posterior means of the GPs. The performance criterion used is the hyper-volume, which is maximized by the actual Pareto set [16]. When the recommendation produced contains an infeasible point, we set the corresponding hyper-volume equal to zero. For each method we report the logarithm of the relative difference between the hyper-volume of the actual Pareto set and the hyper-volume of the recommendation. Figure 1 shows the average results of each method. Note that PESMOC finds better solutions with a smaller number of evaluations. Moreover, BMOO performs worse when $d = 6$, since it provides closer results to those of RS. When the input dimension d grows we have observed that BMOO spends many evaluations of the objectives and the constraints at the corners of the input space $\mathcal{X}$. Finally, we illustrate the utility of the acquisition function of PESMOC on a toy 2-dimensional optimization problem with input domain $\mathcal{X}$ given by the box $[-10, 10] \times [-10, 10]$:

$$\min_{\mathbf{x} \in \mathcal{X}} \quad f_1(x, y) = xy, \quad f_2(x, y) = -yx \quad \text{s.t.} \quad x \geq 0, y \geq 0\,.$$

In this experiment $\mathcal{F}$ is given by the box $[0, 10] \times [0, 10]$. Figure 2 shows the location of first 20 evaluations made by each method. PESMOC and BMOO quickly identify $\mathcal{F}$, and focus on evaluating the functions in that region. By contrast, RS explores the space more uniformly and evaluates the functions more frequently in regions that are infeasible. Figure 2 also shows the level curves of the acquisition function. These functions take high values inside $\mathcal{F}$ and low values outside $\mathcal{F}$.

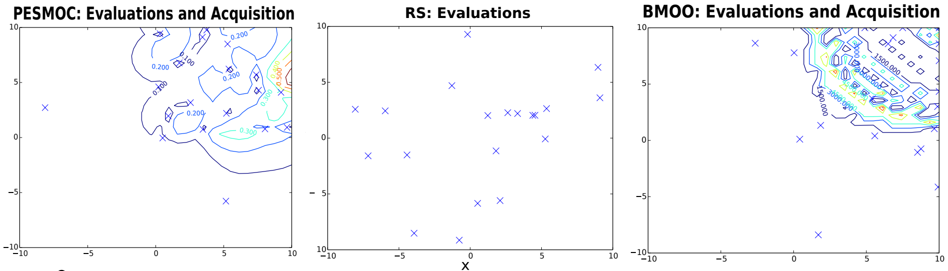

Figure 2: Location in input space (denoted with a blue cross) of each of the evaluations made by PESMOC (left), RS (middle) and BMOO (right). In the case of PESMOC and BMOO, we also plot the level curves of the acquisition function. Best seen in color. The feasible region is the the box $[0, 10] \times [0, 10]$.

4 Conclusions and Future Work

PESMOC is an information-based approach used to address Bayesian optimization problems with multiple objectives and constraints, it evaluates them at a location that is expected to reduce the entropy of the posterior distribution of the Pareto set the most. Synthetic experiments show that PESMOC provides estimates of the Pareto set that are more accurate than a random search strategy and a state-of-the-art method, BMOO. Future work includes considering real-world optimization problems, noisy scenarios, a decoupled evaluation of the functions [7] and a extension to a batch setting in which not only one but a set of points is chosen to evaluate the functions [17].

Acknowledgments

The authors acknowledge the use of the facilities of Centro de Computación Científica (CCC) at Universidad Autónoma de Madrid, and financial support from the Spanish Plan Nacional I+D+i, Grants TIN2013-42351-P and TIN2015-70308-REDT, and from Comunidad de Madrid, Grant S2013/ICE-2845 CASI-CAM-CM.

References

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

[2] P. Siarry and Y. Collette. Multiobjective optimization: principles and case studies, 2003.

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

[4] C. E. Rasmussen. Gaussian processes for machine learning. 2006.

[5] E. Brochu, V. M. Cora, and N. 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.

[6] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104:148–175, 2016.

[7] D. Hernández-Lobato, J. M. Hernández-Lobato, A. Shah, and R. P. Adams. Predictive entropy search for multi-objective Bayesian optimization. In *International Conference on Machine Learning*, 2016. In press.

[8] J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani. Predictive entropy search for Bayesian optimization with unknown constraints. In *International Conference on Machine Learning*, 2015.

[9] 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.

[10] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13:1809–1837, 2012.

[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*, pages 918–926, 2014.

[12] P. Féliot, J. Bect, and E. Vazquez. A Bayesian approach to constrained single-and multi-objective optimization. *Journal of Global Optimization*, 2016. In press.

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

[14] T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369, 2001.

[15] I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems 23*, pages 1732–1740. 2010.

[16] 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.

[17] A. Shah and Z. Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*, pages 3312–3320, 2015.