Learning Stochastic Binary Tasks using Bayesian Optimization with Shared Task Knowledge

Matthew Tesch MTESCH@RI.CMU.EDU
Jeff Schneider JEFF.SCHNEIDER@CS.CMU.EDU
Howie Choset CHOSET@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

Abstract

Robotic systems often have tunable parameters which can affect performance; Bayesian optimization methods provide for efficient parameter optimization, reducing required tests on the robot. This paper addresses Bayesian optimization in the setting where performance is only observed through a stochastic binary outcome – success or failure. We define the stochastic binary optimization problem, present a Bayesian framework using Gaussian processes for classification, adapt the existing expected improvement metric for the binary case, and benchmark its performance. We also exploit problem structure and task similarity to generate principled task priors allowing efficient search for difficult tasks. This method is used to create an adaptive policy for climbing over obstacles of varying heights.

1. Introduction

Many real-world optimization tasks take the form of optimization problems where the number of objective function samples is severely limited. This often occurs with physical systems which are expensive to test, such as choosing optimal parameters for a robot's control policy. In cases where the objective is a continuous real-valued function, the use of Bayesian sequential experiment selection metrics such as *expected improvement* (EI) has lead to efficient optimization of these objectives. An advantage of EI is that it requires no tuning parameters.

We are interested in the problem setting where the objective is not a deterministic continuous-valued function, but a stochastic binary valued function. In the case of a robot, instead of choosing parameters which maximize locomotive speed, the task may be to choose the parameters of a policy which maximize the probability of successfully moving over an obstacle (Fig. 1), where the success of this task is stochastic due to noise in the system.

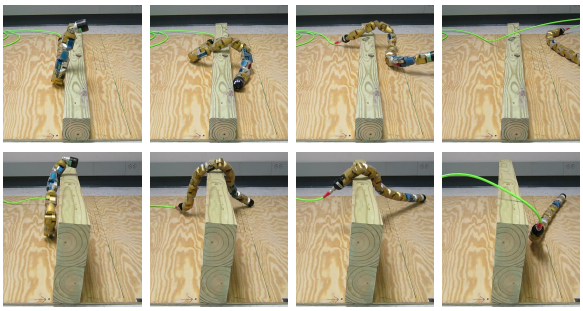


Figure 1. Efficient optimization is possible even with limited function evaluations which only return a noisy 'success' or 'failure'. **Top:** Moving over a 3.5 inch beam with the predicted best motion after 20 evaluations, using no prior information. **Bottom:** Sharing results from previous optimizations on different obstacles allows the robot to move over an 11 inch beam on the first attempt.

Inspired by the success of Bayesian optimization for continuous problems, we propose using a similar framework for the stochastic binary setting. This paper defines the stochastic binary optimization problem, describes the application of Gaussian processes for classification to this problem, proposes a selection metric based on EI, and benchmarks performance on synthetic test functions against a number of potential baseline approaches.

Unfortunately, working in parameter spaces where regions with significant probability of success are relatively sparse (e.g., a snake robot attempting to over-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

come a tall obstacle) will amount to a blind search. Inspired by ideas in multi-task learning, we exploit task structure to solve simpler problems first (smaller obstacles), and then use the learned knowledge as a principled prior for the difficult task. This enables efficient optimization of the task which would otherwise require us to resort to an exhaustive search.

2. Related Work

For optimization problems where each function evaluation is *expensive* (either requiring significant time or resources) the choice of which point to sample becomes more important than the speed at which a sample can be chosen. Thus, Bayesian optimization of such functions relies on a function regression method, such as Gaussian processes (GPs) (Rasmussen & Williams, 2006), to predict the entire unknown objective from limited sampled data. Given this prediction of the true objective, the central challenge is the exploration/exploitation tradeoff – balancing the need to explore unknown areas of the space with the need to refine the knowledge in areas that are known to have high function values. Metrics such as the upper confidence bound (Auer et al., 2002), probability of improvement (Žilinskas, 1992), and expected improvement (Mockus et al., 1978) attempt to trade off these conflicting goals. Existing literature primarily focuses on deterministic, continuous, real-valued functions, rather than stochastic ones or ones with binary outputs.

Active learning (c.f. survey of Settles (2009)) is primarily focused on learning the binary class membership of a set of unlabeled data points but generally attempts to accurately learn class membership of all unlabeled points with high confidence, which is inefficient if the loss function is asymmetric (if it is more important to identify successes than failures). The active binary-classification problem discussed in (Garnett et al., 2012) focuses on finding a Bayesian optimal policy for identifying a particular class, but assumes deterministic class membership.

In the bandit literature, the subtopic of continuous-armed bandits or metric bandits (e.g., (Agrawal, 1995; Auer et al., 2007)) have a similar problem structure to that described in our work; these embed the "arms" of the classic multi-arm bandit problem into a metric space allowing a potentially uncountably infinite number of arms. The focus of much bandit work is minimizing asymptotic bounds on the *cumulative regret* in an online setting, whereas we are concerned only the performance of the algorithm recommendation after an offline training phase.

Prior work in multi-task learning postulates that tabula rasa learning for multiple similar problems is to be avoided, especially when the task has descriptive features (or parameters). Approaches using a number of techniques have been taken (e.g., (Bakker, B. and Heskes, 2003) suggest neural network predictors for generalizing task knowledge), but perhaps the most relevant is that of (Bonilla et al., 2007) or (Tesch et al., 2011); these both incorporate the task as additional parameters of the GP used to model the objective. Bonilla et al. attempt to efficiently and accurately model rather than optimize the objective at a new task. Tesch et al. focus on Bayesian optimization, but allow the algorithm to choose the task parameters of each experiment as well. Additionally, neither of these approaches considers the case of binary information.

3. Binary Stochastic Problem

Given an input (parameter) space $X \subset \mathbb{R}$ and an unknown function $\pi\colon X \to [0,1]$ which represents the underlying binomial probability of success of an experiment, the learner sequentially chooses a series of points $\mathbf{x} = \{x_1, x_2 \dots x_n \,|\, x_i \in X\}$ to evaluate. After choosing each x_i, the learner receives feedback y_i where $y_i = 1$ with probability $\pi(x_i)$ and $y_i = 0$ with probability $1 - \pi(x_i)$. The choice of x_i is made with knowledge of $\{y_1, y_2 \dots y_i - 1\}$. The goal of the learner is to recommend, after n experiments, a point x_r which minimizes the (typically unknown) error, or *simple regret*, $\max_{x \in X} \pi(x) - \pi(x_r)$; this is equivalent to maximizing performance $\pi(x_r)$.

4. Background

4.1. Bayesian Optimization

In Bayesian optimization of a continuous real-valued deterministic function, the goal is to find x_{best} which maximizes the function $f\colon X \to \mathbb{R}$. The process relies on a data-driven probabilistic model $\hat{f}$ (often a GP) of the underlying function f, and a selection metric which selects the next point to sample at each iteration.

The algorithm is an iterative process – at each step i, fit a model based on $\mathbf{x}$ and $\mathbf{y}$, select a next x_i, and evaluate x_i on the true function f to obtain y_i. The crux of the algorithm is the metric which is optimized to choose the next point. EI has been popularized as such a selection metric in the Efficient Global Optimization algorithm (Jones et al., 1998). Given a function estimate $\hat{f}$, improvement is defined as

$$I(\hat{f}(x)) = \max(\hat{f}(x) - y_{best}, 0), \tag{1}$$

where y_{best} was the maximizer of the previously sampled $\mathbf{y}$. The GP defines $\hat{f}(x)$ as a posterior distribution over $f(x)$; the expectation over this, $\text{EI}(x) = \mathbb{E}[I(\hat{f}(x))]$, defines the EI:

$$\begin{aligned}\text{EI}(x) &= (\hat{f}^x_\mu - y_{best})\left(1 - \Phi((y_{best} - \hat{f}^x_\mu)/\hat{f}^x_\sigma)\right) \\ &\quad + \hat{f}^x_\sigma \phi((y_{best} - \hat{f}^x_\mu)/\hat{f}^x_\sigma)\end{aligned}$$

Above, ϕ and Φ are the probability and cumulative density functions (the pdf and cdf) of the standard normal distribution; p^x_f is the pdf at $\hat{f}(x)$, $\hat{f}^x_\mu$, and $\hat{f}^x_\sigma$ are the mean and standard deviation.

4.2. Gaussian Processes for Classification

A key idea behind Bayesian optimization is the probabalistic modeling of the unknown function. In the binary stochastic case, standard GPs are not appropriate because they are a regression technique, fitting continuous data. We use an adaptation of GPs for classification; this provides a similar probabilistic model, but for stochastic binary data (c.f. (Rasmussen & Williams, 2006)).

As in linear binary classification, the use of a sigmoidal *response function* σ[1] converts a model with a range of $(-\infty, \infty)$ to an output that lies within $[0, 1]$ (i.e., a valid probability). In Gaussian processes for classification (GPC), a latent GP $\hat{f}$ defines a Gaussian pdf p^x_f for each $x \in X$ (as well as joint Gaussian pdfs for any set of points in X). The corresponding probability density over class probability functions, p^x_π, is

$$p^x_\pi(y) = p^x_f(\sigma^{-1}(y))\frac{\delta\sigma^{-1}}{\delta y}(y). \tag{2}$$

Although the response function σ maps from the latent space F to the class probability space Π, $p^x_\pi(y) \neq p^x_f(\sigma^{-1}(y))$ due to the change of variables. Also note that as we do not observe values of f directly, the inference step requires an analytically intractable integral. Advantages and disadvantages of different approximate methods are discussed in (Nickisch & Rasmussen, 2008); we use Minka's expectation propagation (EP) method (2001) due to its accuracy and reasonable speed.

[1] In this work, we assume σ is the standard normal cdf; however, any monotonically increasing function mapping from $\mathbb{R}$ to the unit interval can be used.

4.2.1. Expectation of Posterior on Success Probability

As noted above, $p^x_\pi(y) \neq p^x_f(\sigma^{-1}(y))$; therefore the expectation of the posterior over the success probability, $\mathbb{E}[p^x_\pi]$, is not generally equal to $\sigma(\mathbb{E}[p^x_f])$. To calculate the former, we use the definition of expectation along with a change-of-variables substitution ($\pi = \sigma(f)$ and $y = \sigma(z)$) to take this integral in the latent space (where approximations for the standard normal cdf can be used):

$$\begin{aligned}\mathbb{E}[p^x_\pi] &= \int_0^1 y p^x_\pi(y) dy &(3)\\ &= \int_{\sigma^{-1}(0)}^{\sigma^{-1}(1)} \sigma(z) p^x_f(z) \frac{\delta\sigma^{-1}}{\delta y}(\sigma(z)) \frac{\delta\sigma}{\delta z}(z) dz \\ &= \int_{-\infty}^{\infty} \sigma(z) p^x_f(z) dz &(4)\end{aligned}$$

As noted in section 3.9 of (Rasmussen & Williams, 2006), if σ is the Gaussian cdf this can be rewritten as follows (for notational simplicity, we define $\bar{\pi}(x) = \mathbb{E}[p^x_\pi]$ for use later in the paper):

$$\mathbb{E}[p^x_\pi] = \Phi\left(\frac{\mathbb{E}[p^x_f]}{\sqrt{1 + \mathbb{V}[p^x_f]}}\right) \tag{5}$$

5. Expected Improvement for Binary Responses

In the case of stochastic binomial feedback, the notion of improvement that underlies the definition of EI must change. Because the only potential values for y_i are 1 and 0, after the first 1 is sampled y_{best} would be set to 1. Because there is no possibility for a returned value higher than 1, improvement (and therefore EI) would be identically zero for each $x \in X$.

Instead we note that these are noisy observations of an underlying success probability and query the GP posterior at each point in $\mathbf{x}$. Let

$$\hat{\pi}_{max} = \max_{\mathbf{x}} \bar{\pi}(x). \tag{6}$$

As the 0 and 1 responses are samples from a Bernoulli distribution with mean $\pi(x)$, we define the improvement as if we could truly sample the underlying mean. Choosing this rather than conditioning our improvement on 0/1 is consistent with the fact that our $\hat{\pi}_{max}$ represents a probability, not a single sample of 0/1. In this case,

$$I_\pi(\pi(x)) = \max(\pi(x) - \hat{\pi}_{max}, 0) \tag{7}$$

To calculate the EI, we follow a similar procedure to that in §4.2.1 to calculate the expectation of $I_\pi(\pi(x))$:

$$\begin{aligned} \mathrm{EI}_\pi(\hat{\pi}(x)) &= \int_{\hat{\pi}_{max}}^{1} (y - \hat{\pi}_{max}) p_\pi^x(y) dy \\ &= \int_{\sigma^{-1}(\hat{\pi}_{max})}^{\infty} (\sigma(z) - \hat{\pi}_{max}) p_f^x(z) dz \end{aligned} \tag{8}$$

The marginalization trick that allowed us to evaluate this integral and obtain a solution only requiring the Gaussian cdf in the case of $\bar{\pi}$ (Eqn. (5)) does not work because here the integral is not from $-\infty$ to ∞; fortunately it is one dimensional regardless of the dimension of X and is easy to numerically evaluate in practice (e.g., using adaptive quadrature techniques).

6. Performance Benchmarks for Stochastic Binary EI

To validate the performance of our EI metric for stochastic binary outputs, we created several challenging synthetic test functions for $\pi(x)$ on which we could run a large number of optimizations; these functions exhibit properties such as multiple local optima, narrow global optimum, and stochasticity ($\pi(x) \notin \{0, 1\}$ over much of X).

As baselines to compare against the metric we propose in §5, we use uniform random selection, upper confidence bound (UCB) on the latent function $\hat{f}$, EI on the latent function $\hat{f}$ (EI_f), and the Upper Confidence Bound for Continuous-armed bandits algorithm (UCBC) proposed in (Auer et al., 2007). For the β UCB parameter (standard deviation coefficient), we chose the value which performed best, $\beta = 1$, and for the UCBC algorithm we chose the algorithm parameter $n = (T/ln(T))^{1/4} = 2$ via the method given in the paper.[2] Because we are not directly observing the sampled function value, we redefine the y_{best} term for EI_f as $y_{best} = \max_{\mathbf{x}}\{\sigma^{-1}(\bar{\pi}(x))\}$, where $\mathbf{x}$ is all sampled x_i.

To compare the various algorithms, we allowed each algorithm to sequentially choose a series of $\mathbf{x} = \{x_1, x_2 \dots x_{50}\}$, with feedback of y_i generated from a Bernoulli distribution with mean $\pi(x)$ (according to the test function) after each choice of x_i. This was completed 100 times for each test function.[3]

To obtain a measure of the algorithm's performance at step i, we use the natural Bayesian recommendation strategy of choosing the point which has the highest expected probability of success given the predicted function ($x_{best} = \mathrm{argmax}_X \mathbb{E}[p_\pi^x | \{x_1, x_2 \dots x_i\}$ and $\{y_1, y_2 \dots y_i\}]$).[4] The point x_{best} is then evaluated on the underlying true success probability function π, and the resulting value $\pi(x_{best})$ is given as the expected performance of the algorithm at step i. For the random selection and UCBC algorithms which do not have a notion of $\hat{\pi}$, a GP was fit to the data collected by the algorithm to obtain this $\hat{\pi}$ using the same parameters as for the Bayesian optimization algorithms.

In Fig. 2, we plot the average performance over 100 runs of the proposed stochastic binary expected improvement EI_π as well as various baselines. As expected, the knowledge of the underlying function grew slowly but steadily as random sampling characterized the entire function. The focus of EI_π on areas of the function with the highest expectation for improvement led to a more efficient strategy which still chose to explore, but focused experimental evaluations on more promising areas of the search space. Notably, EI_π matched or outperformed tuned versions of all other algorithms tested, *without* requiring a tuning parameter. The UCBC algorithm worked well for simple cases (test function 1 had a significant region with high probability of success) but faltered as the functions became more difficult to optimize; challenges with this algorithm include lack of shared knowledge between nearby intervals, dependence on a tuning parameter (number of intervals), and that it is not defined for higher dimensions.

We also note that EI_π outperforms the naïve use of Bayesian optimization techniques on the latent GP $\hat{f}$, as shown in Fig. 2. This is largely true because the interpretation of variances on the latent function when used in the classification framework are unintuitive – the variance $\hat{f}_\sigma$ is not based solely on the sampled points as in the regression case; instead larger values of $\hat{f}_\mu$ tend to have larger variances due to the nonlinear mapping into the space of probabilities $\hat{\pi}$.

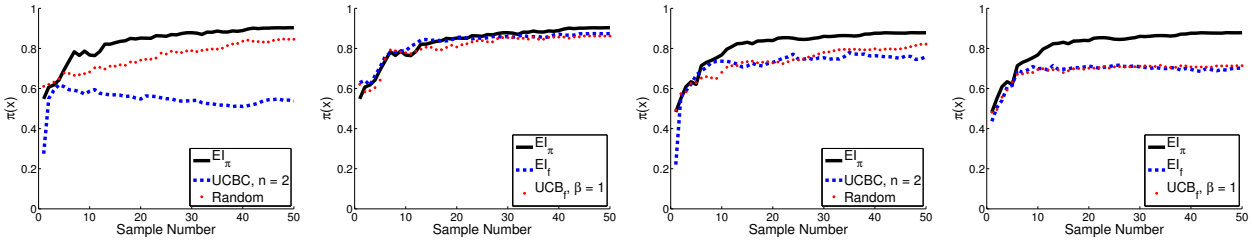
7. Robotic Application: Snakes and Obstacles

The snake robot described in (Wright et al., 2012) has impressive locomotive capabilities, and is able to use cyclic motions called gaits to move quickly across flat

[2] We also set $n = 10$, but obtained similar results.

[3] Our MATLAB implementation of these algorithms and more extensive results are available at `http://www.mtesch.net/ICML2013/`

[4] In practice one may optimize a utility function that considers risk (e.g., the uncertainty in that probability).



(a) Test Function A (b) Test Function A (c) Test Function B (d) Test Function B

Figure 2. After each sample, the algorithm was queried as to its recommendation for a point x that would have a maximum expectation of success $\pi(x)$. These results show the underlying probability value of that point averaged over 100 runs of each algorithm. We compare the stochastic binary EI (EI_π) to the Auer's continuous-armed bandit algorithm UCBC as well as uniform random selection in (a) and (c), and to EI and UCB on the latent function in (b) and (d). More complete results are available online.

ground, forward through narrow openings, and up the inside and outside of irregular horizontal and vertical pipes. However, moving over cluttered, obstacle laden surfaces (such as a rubble pile) provide a challenge for the system. One such obstacle encountered in field deployments is a 4x4 beam., as we have encountered in the field during disaster response training exercises.

A master-slave system was set up to record an expert's input to move the robot over the obstacle. Using a sparse function approximation of the expert's input, we created a 7-parameter model that was able to overcome obstacles of various sizes, albeit unreliably – the same parameters would only sometimes result in success. Parameters of this model (offset, widths, and amplitudes of travelling waves) were difficult to optimize by hand to produce reliable results.

Using the EI_π metric, a 3-dimensional subspace was searched to identify parameters which resulted in a robust motion over the original obstacle. Running 20 robot experiments (function evaluations) resulted in the recommendation of a parameter setting which produced robust, successful motions (top of Fig. 1).

Attempting this same optimization on a 9 inch obstacle resulted in no successes within the first 20 trials; a solution with a non-zero probability of success was sparse enough that we were essentially conducting a blind search of the parameter space.

7.1. Exploiting Task Structure to Solve Difficult Problems

We wish to avoid an exhaustive search, even for problems where the regions with high success probability are sparse within the space. When these problems represent the optimization of a task, such as a robot moving over an obstacle, one can often *parameterize* that task. With a carefully chosen task parameterization, one can learn the general behavior and location of optima of the objective from one of more simpler optimization problems, and use these as a principled prior for optimization of the difficult task.

Applying ideas from (Bonilla et al., 2007) and (Tesch et al., 2011) to the snake robot task, we attempted to learn parameters of our expert-based model for more difficult obstacles, such as the 9 inch beam we could not overcome. We added a fourth parameter, representing obstacle height, to our GP function approximation. This generated a prediction for all obstacle heights, allowing us to have a strong prior for subsequent optimizations by incorporating previous data. Figure 3 shows a selected trial for each intermediate task parameter (5.5, 7, and 9 inches), each using the data from all previous optimizations.

As opposed to the initial experimental trial, we found a successful 9 inch trial on the first experiment suggested by EI_π, demonstrating that shared knowledge between tasks can improve real-world optimization performance. Parameters for overcoming an 11 inch beam were then successfully predicted with no required optimization (Fig. 1).

Although generalizing results from an easier task to a more difficult task works well for many problems, there are caveats. Common choices for GP covariance functions are axis-aligned, resulting in poorer generalization if a trend across multiple tasks exists with principal direction that is not primarily along the task parameter axes. In addition, if a global optima for a difficult task is unrelated to an optima for a simple task, the sharing of knowledge across tasks is less likely to increase efficiency (unless it helps identify global properties of this function that could improve the search).

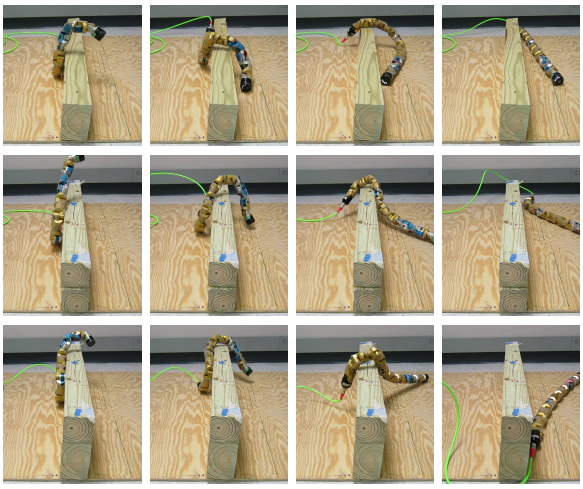


Figure 3. Successful trials from optimizations for 5.5 (top), 7 (center), and 9 (bottom) inch obstacles. The robot started at the left side of the obstacle, and moved over the obstacle to the right.

8. Conclusion and Future work

We have defined the stochastic binary optimization problem for expensive functions, presented a novel use of GPC to frame this problem as Bayesian optimization, and presented a new optimization algorithm that computes expected improvement in the stochastic binary case, outperforming several baseline metrics as well as a leading continuous-armed bandit algorithm. We used our algorithm to learn a robust motion for moving a snake robot over an obstacle, and used multi-task learning concepts to efficiently create an adaptive policy for obstacles of various heights.

The problem we define is not limited to the demonstrated snake robot application, but applies to many expensive problems with parameterized policies and stochastic success/failure feedback, including variations of applications where continuous-armed bandits are currently used such as auction mechanisms and oblivious routing which could contain an offline training phase penalizing simple rather than continuous regret.

References

Agrawal, Rajeev. The Continuum-Armed Bandit Problem. *SIAM Journal on Control and Optimization*, 33(6):1926–1951, November 1995. ISSN 0363-0129. doi: 10.1137/S0363012992237273.

Auer, Peter, Cesa-Bianchi, N, and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, pp. 235–256, 2002.

Auer, Peter, Ortner, Ronald, and Szepesvári, C. Improved rates for the stochastic continuum-armed bandit problem. *Learning Theory*, 2007.

Bakker, B. and Heskes, T. Task Clustering and Gating for Bayesian Multitask Learning. *Journal of Machine Learning Research*, (4):83–99, 2003.

Bonilla, E, Agakov, F, and Williams, C. Kernel multi-task learning using task-specific features. In *11th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

Garnett, Roman, Krishnamurthy, Yamuna, Xiong, Xuehan, Schneider, Jeff, and Mann, Richard P. Bayesian optimal active search and surveying. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, 2012.

Jones, Donald R., Schonlau, Matthias, and Welch, William J. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4), 1998. ISSN 0925-5001.

Minka, Thomas P. *A family of algorithms for approximate Bayesian inference*. Phd thesis, Massachusetts Institute of Technology, 2001.

Mockus, J, Tiesis, V, and Zilinskas, A. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

Nickisch, Hannes and Rasmussen, CE. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008.

Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 026218253X.

Settles, Burr. Active Learning Literature Survey. Technical report, University of Wisconsin–Madison, 2009.

Tesch, Matthew, Schneider, Jeff, and Choset, Howie. Adapting Control Policies for Expensive Systems to Changing Environments. In *International Conference on Intelligent Robots and Systems*, 2011.

Žilinskas, Antanas. A review of statistical models for global optimization. *Journal of Global Optimization*, 2(2):145–153, June 1992. ISSN 0925-5001.

Wright, C., Buchan, A., Brown, B., Geist, J., Schwerin, M., Rollinson, D., Tesch, M., and Choset, H. Design and Architecture of the Unified Modular Snake Robot. In *2012 IEEE International Conference on Robotics and Automation*, St. Paul, MN, 2012.